

Audion Office OCR AI - User Guide

Contents

- [Quick Route](#)
- [The Window](#)
- [Source and Target](#)
- [Sections](#)
- [Deliverables](#)
- [Scan Cleanup](#)
- [Models and Keys](#)
- [Gemini Standard, Flex and Batch](#)
- [Main Folders](#)
- [The Console Command](#)
- [Before Large Runs](#)
- [Privacy](#)

Audion Office OCR AI converts office files, PDFs, images and scans into a provider-neutral

DocumentModel, then immediately writes the selected office, archival and machine-readable results.

Markdown is an optional audit/LLM export, not the center of the system.

Quick Route

1. Start **Start.exe**.
2. Put the files into **input**, or choose another folder or one file as **SOURCE** in the right column.
3. Open a section in the band under the header, then its page, then the switch of the engine or the provider.
4. Tick the deliverables, press the run button at the right of the page head.
5. Watch the terminal; the results are in **output**, or in the folder chosen as **TARGET**.

The Window

The header names the machine and what it has for OCR: Windows, the computer, the processor, the video card with its driver, the memory, the drives, the recommendation for the video card, Tesseract with its languages, the layout service of Surya, and the providers that have a key. Every badge says the whole of it in its tooltip. **RU** / **EN** switch the language, the sun switches the theme. **SETUP** opens the installation of what the machine needs for local OCR (Tesseract, Real-ESRGAN, llama.cpp and Surya runtimes - the header recommends the one for this video card) - start there: while the machine has no Tesseract, the window opens on **SETUP** at start. **SETTINGS** sets the fonts and their sizes, **ABOUT** names the version and the folders.

Under the header the sections stand in one band: **LOCAL OCR**, **API OCR**, **OCR CHECK**, **TABLES**, **TOOLS**. A section with several pages shows them as a second band. A page of OCR has a third band - the engines of **LOCAL OCR**, the providers of **API OCR**; the switch that is on fixes the engine of the form.

A form stands in blocks with their names; a coloured stripe at the left of a block says what kind of settings it holds. Settings that are rarely needed stand under **ADDITIONAL**, closed until it is opened. The arrow at the right of the page head puts the fields of the page back to the values of the program. A switch of a risky choice is drawn red.

The right column holds the working area - **SOURCE** and **TARGET** - and the terminal. The terminal shows the line every run is started with, what the operation says in its colours, and one live line of what it is doing now. **STOP** asks the operation to stop at its next safe point; a second press a few seconds later ends it at once. Closing the window while an operation goes asks first.

Source and Target

- **SOURCE** is a folder or one file. **input** of the program is read as it is. A source outside the program is copied into **input** before a run, and **input** and **output** are cleared first; when they hold anything, the window asks before the run and says how many items each holds. A source inside **output** and a folder inside **input** are refused before anything runs.
- **TARGET** is the folder the contents of **output** are copied into after a run.
- A folder or a file dropped on the window becomes the source. The clock buttons open the history of paths, the pin keeps a path on top of it, **RESET** sets both paths back to **input** and **output**.
- **LIST** names in the terminal what lies in the source.

- The bin button opens the clearing: the contents of the source, of the target or of both - into the Recycle Bin, the folders stay; a folder outside the program is asked about once more. The same menu clears the DocumentModel artifacts.
- The arrows button copies the Markdown of `output\` back into `input\`.

Sections

`TOOLS` -> `To Markdown` extracts selectable text from DOCX, XLSX, PPTX, PDF, TXT, CSV and HTML without OCR.

`LOCAL OCR` runs free local OCR. Tesseract is the recommended baseline. Surya is the optional slow local engine with the layout of the page; under its switch a line says whether its layout service runs, with the button that starts or stops it. The service keeps the layout model in memory between runs.

`API OCR` runs Yandex, Mistral OCR, xAI, Gemini and OpenAI vision models. The switch of the provider shows its fields: the model, the key, the parameters. Mistral supports a managed second pass: `None`, `Tesseract`, `Yandex`, or `Yandex + Tesseract`. Gemini can use `Standard` or `Flex`; Flex is cheaper, but may wait longer for available capacity.

`OCR CHECK` holds `Compare engines` - one page by several engines, raw and cleaned, side by side before a large batch - and `Check the numbers` - numeric and legal lines of one page read again by a vision model and compared.

`TABLES` holds three operations, each with its settings and its button: coordinate tables of a scan into XLSX, the check of Markdown tables, Markdown tables into XLSX. For ruled scans the OCR also recovers a physical grid, validates its column count and compares provider text inside verified cells.

`TOOLS` holds `To Markdown` (office files quickly, without OCR), `Checks` (the project, the environment, the models and the keys of the providers, short OCR tests), `Workbench` (the local review of weak OCR, the AI resolver of doubtful crops, coordinates), `Rebuild Markdown` (existing Markdown into DOCX, LLM-DOCX, PDF, PPTX - compatibility, not the main OCR path) and `Gemini Batch`.

Deliverables

Both OCR pages show the `Deliverables` block first. DOCX is the working default. Four office formats (`DOCX`, `XLSX`, `Searchable PDF`, `ODT`) and four audit formats (`Markdown`, `OCR JSON`, `HTML`, `Verification`) are independent and can be produced together. Every selected result is written from the saved `DocumentModel` without running OCR again.

Each `<name>.document` package is the canonical internal OCR record. It retains the source, page images, OCR candidates, coordinates, tables, reading order, confidence, verification and manual corrections. Completeness means preservation of source data, not a promise of perfect recognition accuracy.

The `Output` block presents `Text + boxes` and `Layout`. This contract controls OCR data fidelity, not the final office format.

Scan Cleanup

The `Scan cleanup` block holds one row of profiles:

- `Auto` uses tuned defaults for the selected engine.
- `Raw` sends the page unchanged.
- `Heavy scan` boosts denoise, contrast and sharpening.
- `Numbers` is tuned for legal and financial identifiers.
- `Manual` opens the detailed preprocessing controls under `ADDITIONAL`.

Aggressive preprocessing can erase punctuation, thin table lines, stamps or small requisites. Keep the original page and compare a cleaned sample at full resolution.

Models and Keys

A list of models opens under its field. The program keeps the lists it has read; `REFRESH THE LIST` asks the provider again by its key.

API keys live in `config\`. The `+` beside the key field adds a key: it is pasted into a dialog, written into a new file of `config\` and becomes the key of the field; the window and the journal never show it. The bin beside it sets the chosen key file aside into `config\deleted_keys\`, from where it can be brought back. Do not include real keys in public archives.

Gemini Standard, Flex and Batch

`Standard` is the normal interactive Gemini request. Use it for one-page checks, OCR comparisons and manual experiments.

`Flex` is an interactive request with `service_tier=flex`. It is cheaper, but may wait longer for available capacity. The form names it `Gemini tier`; Flex uses a longer timeout.

Batch is the asynchronous mode for large non-urgent OCR runs. **Gemini Batch** prepares `workspace\gemini_batch\...\requests.jsonl` and `manifest.json` by default; the paid Google submission happens only when **Submit to Google** is enabled. The collect operation checks the job and downloads Markdown. Details are in `docs\tools\GEMINI_BATCH_FLEX.md`.

For literal xAI OCR, try **4.20 fast** (`grok-4.20-non-reasoning-latest`) first. **4.20 quality** (`grok-4.20-reasoning-latest`) is the slower option for difficult tables and fragmented scans. **Russian OCR review** is off by default; when enabled, the reasoning model reviews only suspicious pages.

Main Folders

<code>input\</code>	source documents
<code>output\</code>	user-facing deliverables
<code>config\</code>	settings, prompts and API keys
<code>docs\</code>	project documentation
<code>docs\PDF\</code>	PDF documentation
<code>logs\</code>	operation logs
<code>report\</code>	service reports
<code>workspace\</code>	temporary working files and checks
<code>data\</code>	the lists of models the program keeps
<code>cache\</code>	OCR/preprocess cache
<code>App\</code>	the window and the PowerShell of the program

The Console Command

Every operation of the window runs from a console through `ocr_cli.cmd`, with the same settings:

```
ocr_cli.cmd describe
ocr_cli.cmd run --operation ocr_local_run --source D:\Scans --target D:\Results --set
ocr_engine=tesseract
ocr_cli.cmd options --operation ocr_api_run --field gemini_model --refresh
ocr_cli.cmd hardware
ocr_cli.cmd version
```

describe prints the sections, operations and fields as JSON; **run** runs one operation; **options** prints the list of a field (`--refresh` asks the provider); **key add|remove** adds a key read from the standard input or sets a key file aside; **hardware** names the video card, Tesseract and the layout

service. Exit codes: 0 done, 1 failed, 130 stopped, 2 a wrong call - refused before anything runs. The window writes the line it runs into its terminal at every start.

Before Large Runs

Run `Compare engines` on one page first. For legal documents, also run `Check the numbers`. Then choose the engine, the cleanup profile and the deliverables. Review DOCX, XLSX and the `report` artifacts after OCR; Markdown is needed only when selected as a separate audit export.

Use local OCR when documents must stay offline and the installed runtime supports the language and the layout. Use a paid OCR API when its document and layout capabilities justify external processing and the data policy permits upload.

Check row and column boundaries, merged cells, numeric separators, dates, identifiers, organization names, signatures and stamps. `Check the numbers` is a targeted validation step and does not replace review of the full recognized document.

Before a large batch, verify the source count, the target, the provider and model, the API quota and the free disk space. Keep failed source pages and their logs together; retry only failed documents.

Privacy

Treat scans, recognized text, structured exports, logs and provider requests as sensitive. With `API` `OCR` and the AI resolvers the pages go to the chosen provider with its key. API keys stay in `config\` and are never copied into reports or documentation.