

# Audion Image Tools - User Guide

---

## Contents

- [Purpose](#)
- [Quick Route](#)
- [New GUI Layout](#)
- [Workbench I/O](#)
- [Modules](#)
- [Maintenance](#)
- [Main Folders](#)
- [CLI Examples](#)
- [Before Release](#)
- [Canonical Workbench labels](#)

This is the English companion to `USER_GUIDE_RU.md`.

## Purpose

---

Audion Image Tools is a portable GUI/CLI toolkit for batch image and PDF processing: conversion, PDF image extraction, image-to-PDF, crop, resize, DPI metadata, color profiles, printable sheets, contact sheets, and diagnostics.

Primary launch:

```
launcher_gui.cmd
```

## Quick Route

---

1. Put files into `input\`, stage them from the GUI, or choose a folder in `SOURCE`.
2. Open `launcher_gui.cmd`.

3. Choose a ROOT module: **Conversion**, **PDF**, **Crop**, **Resize**, **Color**, **Photo sheet**, **Sheets and labels**, **Contact sheet**, or **Diagnostics**.
4. Use the action grid at the top of the module. Quick actions run immediately; parameterized actions show their settings in the same window.
5. Check **SOURCE**, **DESTINATION**, and the visible settings.
6. Click the amber **RUN** button when the action has parameters.
7. Read results from **output\** or from the selected **DESTINATION**.

Empty **SOURCE** and **DESTINATION** fields use command defaults, usually **input\** and a subfolder under **output\**.

## New GUI Layout

---

The left navigation contains ROOT modules only. **Conversion** opens directly as a working window. Other modules show action buttons at the top and the selected action settings below.

Action buttons and **RUN** use a muted amber highlight with light gray text. Navigation buttons remain standard blue Quasar buttons. Settings are grouped into source, output/result, format, parameters, and options blocks.

Short labels are intentional. Longer explanations live in tooltips on headings, buttons, checkboxes, radio controls, and toggles. The tooltip background is **RGB(23, 33, 43)**. The **BACK** button has no tooltip.

## Workbench I/O

---

The two top rows define the active processing route:

- **Source** can be the internal **input\** folder, an external folder, or a single external file.
- **Target** can be the internal **output\** folder or a chosen external folder.
- The system folder picker sits at the right edge of each row.
- The pin button shows the route state: an active pin is highlighted, an inactive one is muted. Pinning saves the path to the cache but does not make it the mandatory route for a run.
- The delete button on the left clears the contents of the current folder without removing the folder itself. The internal **input\** needs no confirmation; an external source raises a warning. If the source is a single file, deleting that file is confirmed separately — it may be the only copy.

The bottom row is split into groups by meaning:

| button      | what it does                                                                                                                        |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Source      | opens the current source in the file manager; for a single file, its folder with the file selected                                  |
| Add file... | picks exactly one file without copying; the source path becomes that file until the route changes                                   |
| Target      | opens the current output folder                                                                                                     |
| Reset       | drops every unpinned path from the cache and returns the active routes to the internal <code>input\</code> and <code>output\</code> |
| Delete      | after one shared confirmation, clears the contents of the current source and target                                                 |
| List        | shows the files of the current source; a single file appears as one line                                                            |

**Reset** deletes neither files nor pinned entries — it is the safe way to clear routes before running the project on another machine.

A long path is truncated only on real overflow and fades softly in a narrow zone at the right edge; a short path is shown in full. A path change is marked by a brief highlight, and these effects switch off with the system reduced-motion setting.

## Modules

### Conversion

The Conversion window contains quick actions at the top and the normal conversion settings below.

Quick actions:

- `JPG 75% quality`
- `PNG`
- `Grayscale PNG`
- `2160p JPG 90% quality`
- `16:9 JPG 90% quality`

`Convert images` supports `JPG`, `PNG`, `TIFF`, `WebP`, `AVIF`, `HEIF`, and `HEIC`, with 60/75/90 presets and an exact 1..100 quality field defaulting to 83. DPI writing is available for `JPG`, `PNG`, and `TIFF`.

The **2160p JPG 90% quality** preset uses Lanczos. When the source is actually upscaled to UHD, a mild **Unsharp Mask** is applied after resizing. Downscale to UHD does not add extra sharpening.

## PDF

Actions:

- **PDF export** - export pages by embedded images or full render.
- **Extract from PDF** - extract embedded raster objects, defaulting to original format.
- **Images to PDF** - build PDF files from images.

**PDF export** defaults to embedded raster extraction when possible. If the full page appearance is needed, choose render mode and set DPI. Output format and quality are controlled in the **Format** block.

**PDF rasterize** renders full PDF pages into images at 150/300/600 DPI. Use it when the PDF page itself matters, not only the embedded raster objects inside it.

**Extract from PDF** defaults to writing embedded images in their original format. You can also convert them to target formats, choose flat output or per-PDF folders, and optionally write TIFF sidecars next to PNG.

**Images to PDF** defaults to separate PDFs by folder, preserved source DPI, and lossless PNG PDF mode. A custom PDF DPI becomes active only when DPI preservation is off.

## Crop

Actions:

- **Trim border** - remove a solid frame using the corner color.
- **Smart white crop** - trim white background to meaningful pixels.
- **Split...** - split images into strips or rectangular grid cells.

**Safety margin, mm = 0** cuts exactly to the detected content. A value above zero keeps source background around the result using image DPI; if DPI is missing, 300 DPI is used.

**Trim border** and **Smart white crop** default to PNG, but output can be changed to **JPG**, **TIFF**, **WebP**, **AVIF**, **HEIF**, or **HEIC**. Crop actions can also build a PDF after cropping with embedded PNG or embedded JPG 75% parts without changing pixel resolution.

**Split...** writes parts as **name\_01.png**, **name\_02.png**, and so on. **Strips** uses direction and part count; **Grid** uses row and column counts.

## Resize

Actions:

- **Screen** - 1080p, 1440p, or 2160p/UHD.
- **Aspect** - 16:9, A4, or A3.
- **Downscale** - 25%, 50%, a custom percent, or automatic fit to a target file size in MB.
- **Roll** - write DPI for a target physical side without changing pixels.
- **DPI** - write DPI only.

**Screen -> 2160p** applies a mild Unsharp Mask only on real upscale. **Downscale** applies a mild Unsharp Mask after Lanczos by default. Scale modes are **25%**, **50%**, **Custom %**, and **Fit**. **Fit** searches for the largest percent that keeps each output file below the target MB value; the target spinner steps by **0.1 MB**, and **0.8-1.0** of the target is treated as a good hit.

Resize algorithms:

- **Lanczos** - best visual quality for photos and soft images, but it can create halos and inflate PNG maps.
- **Bicubic** - good photo/JPG compromise: smoother than simple averaging, but it can add intermediate colors around sharp map lines.
- **Box** - dry averaging filter for maps, plans, screenshots, and flat graphics; usually fewer halos and smaller PNG than **Lanczos** / **Bicubic**.
- **Nearest** - creates no new colors and is often the smallest PNG option, but diagonals, labels, and thin lines can become jagged.

Practical rule: start with **Lanczos** or **Bicubic** for photos/JPG; for PNG maps and plans, use **Nearest** for minimum size or **Box** for a calmer visual compromise.

Tooltip details:

- **Fit** does not change JPG quality by itself. It searches for a resize percent that fits the target file size. For JPG, the final size depends on scale, quality, and image complexity; for PNG, it depends on color count, sharp edges, and the resize algorithm.
- **Target MB** is a per-file limit. If the target is **1.0**, the engine tries to land close to 1 MB, but exact byte-level matching is not expected; about **0.8-1.0** of the target is treated as a good result.
- **Lanczos** is attractive for photos and soft scans, but on maps it can create semi-transparent transitions around hard lines. That often increases PNG size.

- **Bicubic** is useful when you want a softer result than **Box**/**Nearest** without the full cost and halo tendency of **Lanczos**.
- **Box** is usually the safest first try for maps, plans, diagrams, and screenshots: it averages pixels simply and tends to create fewer halos.
- **Nearest** is useful when original colors and minimum PNG size matter, but it does not smooth diagonals or small text.

## Color

Normalize to keep, sRGB, or CMYK, then choose output formats, quality, and metadata policy.

## Photo Sheet

Build printable sheets. **Paper saver** creates dense rows without gaps. **Cut-lines** groups similar heights and writes **cut\_plan.txt**. By default, source pixels are not resampled: sheet width and DPI describe the final output sheet. If the selected DPI is too low for the unchanged source pixels at the selected sheet width, the engine raises the sheet DPI automatically and reports a warning in the GUI/report.

## Sheets And Labels

**Tile sheet** repeats one image on A5/A4/A3 or a roll. **Split TIFF** exports multi-page TIFF frames to PNG. **Watermark** adds either a horizontal corner label or a diagonal protective text mark.

**Watermark** has two modes. **Corner** places normal horizontal text into the selected corner with a controlled margin and font size. **Diagonal protective** centers text across the image diagonal and auto-fits the font size to the selected diagonal coverage; **60%** is a practical default for protective document marking and leaves room for short large words such as **COPY**. The font is intentionally simple and readable: Arial Bold when available. Opacity is adjustable separately: **0** is invisible, **255** is fully opaque. Default **64** is about 25% opacity and is chosen for print, where watermarks usually look darker than on screen.

Watermark tooltip details:

- **Corner** is for tidy technical marking: label, date, author, or draft status.
- **Diagonal protective** is for discouraging misuse: the text crosses the document and remains visible even after partial cropping.
- **Diagonal coverage, %** controls target text length relative to the image diagonal, not the font size directly. Shorter text becomes larger at the same coverage percent.

- **Corner font size** applies only to **Corner**. In diagonal mode the font size is calculated automatically.
- **Opacity** is the **0..255** alpha channel. Default **64** is about 25% opacity. For red or black diagonal text, start around **64-96**; for light text on dark images, start around **96-128**.
- **Color** in the GUI is selected from human presets with visible swatches: **Red**, **Black**, **White**, **Yellow**, **Gray**, **Blue**. For exact colors, use **Custom RGB** with **R**, **G**, and **B** channels from **0** to **255**. CLI users can still pass an HTML color or RGB manually, for example **#cc0000** or **204,0,0**.

## Contact Sheet

Create an inspection grid with filename, **px | cm | DPI**, and KB under each thumbnail. The thumbnail can be chosen from a list or entered manually.

## Diagnostics And Maintenance

Diagnostics contains **Formats**, **Information**, and **Doctor**. Maintenance actions are separate; global multithreading defaults live there, and **Clean I/O** clears managed **input\** and **output\** folders with confirmation.

## Multithreading And RAM

Multithreading processes independent source images in parallel. It helps with batches, but large maps, TIFF/PDF rendering, and PNG output can occupy far more memory than the source file size on disk. Each extra worker can open another heavy image, so speedups can turn into paging or out-of-memory failures.

Practical guide:

- **16 GB RAM** - keep multithreading disabled; use **1-2** workers only for small files.
- **32 GB RAM** - usually **4** workers.
- **64 GB RAM** - **8-12** workers after checking peak memory.
- **128 GB+ RAM** - **16+** can be tested, but memory must be watched.

The global Maintenance setting is the default for modes without a local override. A local mode switch affects only the current run and does not change Workbench/I/O.

## Maintenance

The maintenance block sits apart from the working commands. It holds the global threading setting. Route and file handling live in the Workbench above: **Reset** clears routes and cache without deleting files, **Delete** clears the contents of the current source and target after confirmation.

## Threading and memory

Threading processes independent images in parallel. It helps on batches, but large maps, TIFF and PDF rendering, and PNG output can occupy far more memory than the file takes on disk. Every extra worker may open another heavy image, so the speed-up turns into swapping — or a crash — sooner than expected.

A practical guide:

| memory           | threads                                 |
|------------------|-----------------------------------------|
| 16 GB            | leave threading off; 1–2 if needed      |
| 32 GB            | usually 4                               |
| 64 GB            | 8–12, after checking peak memory        |
| 128 GB and above | 16 and up, but only with memory watched |

The global setting acts as the default for modes without a local override. A local switch inside a mode affects only the current run and does not change the Workbench routes.

## Main Folders

```
input\      source files
output\     processed results
logs\       operation logs
report\     GUI reports
workspace\  temporary work area
config\     manifest, GUI settings, themes, ICC profiles
docs\       documentation
system_core\ GUI/CLI code
```



## CLI Examples

---

Use the project runtime:

```
runtime\python.exe system_core\main.py --help
runtime\python.exe system_core\main.py convert --help
runtime\python.exe system_core\main.py resize-screen --help
runtime\python.exe system_core\main.py images-to-pdf --help
```

## Before Release

---

Run compile checks, `system_core\doctor.py`, and the GUI smoke test described in `RELEASE_GUIDE_RU.md` / `RELEASE_GUIDE_EN.md`.

## Canonical Workbench labels

---

Workbench uses the same Audion Image Tools public vocabulary in every project. Its buttons always keep the same order and labels: **Source**, **Add file...**, **Target**, **Reset**, **Delete**, **List**.

**Reset** returns to project `input/output` and does not delete files; **Delete** clears the current **Source** and **Target** only after confirmation. The exact Russian labels are **Источник**, **Добавить файл...**, **Назначение**, **Сбросить**, **Удалить**, **Список**. The Workbench variants **Destination**, **Clear**, **Цель**, and **Очистить** are not used.