imPRESS Studio

Complete technical & user documentation

Software version: 1.5.12

Platform: .NET 8 · Avalonia 11 · Windows x64

Author: Wojciech Bujacz

Document revision: 2026-08-31

Table of contents

1. Introduction

imPRESS Studio is a professional PDF imposition tool for print shops, DTP studios and designers preparing publications for offset, digital and Print-on-Demand production.

Imposition is the process of arranging document pages on press sheets in the order and orientation needed so that, after printing, folding and trimming, a finished publication with correct pagination results. imPRESS Studio automates this: it takes PDF files, lays them out on press sheets (SRA3, A3, B2, Letter or any custom size) according to the chosen binding method, generates printer marks, compensates for creep, and exports a production-ready file — optionally as PDF/X-4.

1.1. Who it is for

- **Offset and digital print shops** — preparing forms, signatures, saddle-stitch and perfect-bound jobs.
- **DTP studios and designers** — quickly checking how a publication tiles onto a sheet and preparing print-ready files.
- **Label and sticker production** — step-and-repeat, roll-fed, Summa OPOS contour-cut markers.
- **Print-on-Demand operators** — automation via hot folders and the command-line interface.

1.2. Key capabilities

Area	Features
Bindings & layouts	Saddle stitch, perfect bound, wire/spiral, N-up ganging, Z-fold, gatefold, accordion, roll-fed step-and-repeat, sheet-fed step & repeat
Marks	Crop marks, registration marks, CMYK colour bars, K grayscale wedge, fold marks, barcodes, Summa OPOS / OPOS-XY contour markers, reusable marks profiles
Prepress	Creep compensation, overprint simulation (multiply blend), bleed, safe zone, spine
Quality control	Deep preflight (image DPI, RGB/CMYK, font embedding, transparency, spot colours, PDF/X compliance), preflight report
Export	Multi-sheet PDF (version 1.4–1.7 selectable), PDF/X-4 (ISO 15930-7) with an ICC profile — natively via the imPRESS Export Engine (including vector-RGB conversion through ICC) or via Ghostscript; PDF/X-1a and X-3 via Ghostscript; page numbering, cutting-guide page
Automation	CLI (impose / license / hotfolder), hot folder subsystem with queue, SQLite ledger and Ghostscript throttling
Productivity	Interactive preview with thumbnails, material-usage and cost statistics, logo overlay, template suggestions, multi-language localization

2. System architecture

imPRESS Studio uses a layered architecture with a clear separation between domain logic, PDF processing, export, automation and presentation. All projects compile into a single executable (`imPRESS Studio.exe`) — the app project includes the other modules' source files via `<Compile Include>`, which simplifies distribution and avoids DLL locking.

2.1. Modules (projects)

Module	Responsibility
Imposition.Core	Imposition domain logic: the template model (`Template`), job (`ImpositionJob`), engine (`ImpositionEngine`), binding strategies, sheet layouts, marks, overlay, layout optimizer, template-suggestion engine, job statistics.
Imposition.Pdf	PDF operations: analyzer (PdfAnalyzer), composer (PdfComposer), exporter (PdfExporter), preview renderer (PDFium), marks and overlay renderers, PDF merging, preflight models and parsers (fonts, images, transparency, spot colours, standard compliance).
Imposition.Export	The export pipeline (`ExportPipeline`), preflight validator, ICC profile manager, PDF/X definition file generation (`PdfXDefinitionFile`), the prepress validation engine with a validator set, the summary generator, range expressions (RangeExpression).
Imposition.HotFolder	Automation subsystem: configuration, file watcher (FileSystemWatcher / polling), event debouncer, file-stability probe, job queue, backoff policy, SQLite ledger, Ghostscript throttle, host and manager.
Imposition.Ui	Avalonia presentation layer: main window, controls (incl. ImpositionCanvas), dialogs (export, templates, hot folders, preflight, license, settings, help), view-models (MVVM, CommunityToolkit.Mvvm), markup extensions (localization).
Imposition.Utils	Shared utilities: app paths (AppPaths), file ops, undo/redo stack, logger setup (Serilog), licensing (machine fingerprint, license file, generator, validator), localization, display units.
Imposition.App	Entry point: `Program.cs` (GUI/CLI routing), `CompositionRoot.cs` (DI container), `CliRunner.cs` (batch commands).

2.2. Technology stack

Component	Technology	Role
Runtime	.NET 8 (net8.0)	Target platform, self-contained win-x64
UI	Avalonia 11.2.7 (Fluent, Skia)	Cross-platform UI, rendering via SkiaSharp
UI pattern	CommunityToolkit.Mvvm 8.2.2	MVVM, generated properties and commands
PDF analysis	PdfPig 0.1.9	Reading pages, fonts, images, boxes
PDF composition	PdfSharpCore 1.3.65	Placing pages on sheets, drawing marks
Preview rendering	Docnet.Core 2.6.0 (PDFium)	Raster sheet preview
Graphics	SkiaSharp 2.88.9	Drawing preview and overlay
Barcodes	ZXing.Net 0.16.10	Generating barcodes on marks
PDF/X-4	imPRESS Export Engine (native) or Ghostscript 10.07 (bundled)	Packaging to the print standard with ICC; natively also vector-RGB conversion (WCS)
State store	Microsoft.Data.Sqlite 8.0.10	Hot folder ledger (idempotency)
Hosting/automation	Microsoft.Extensions.Hosting, Polly 8.4.2	Hot folder hosting, retry/backoff
CLI	System.CommandLine 2.0 beta	Batch command parser
Logging	Serilog 3.1.1 (Console, File)	Structured logs
Hardware/license	System.Management 8.0.0	Machine fingerprint

2.3. Dependency composition & the STA thread

`CompositionRoot.Build()` builds a single `IServiceProvider` shared by the GUI and CLI paths. Long-lived services (engine, analyzer, composer, validators, ICC profile manager, hot folder subsystem) are registered as **singletons**; view-models as **transient** (each window gets a fresh instance). The exception is `TemplateEditorViewModel` — a singleton, so in-progress template edits survive across dialog opens.

> **STA thread:** `Main` is deliberately synchronous and marked `[STAThread]`. The STA attribute only "sticks" to the entry method and is lost after the first `await`; Win32 drag&drop registration (`RegisterDragDrop`) requires the thread to be in STA mode. The GUI therefore starts synchronously, while the CLI path (which does not need STA) is delegated to an async helper.

All binding strategies are registered as `IBindingStrategy` and injected into `ImpositionEngine` as a dictionary keyed by binding type. All preflight validators are registered as `IPrepressValidator` — registration order defines report order.

2.4. Data flow (from file to press)

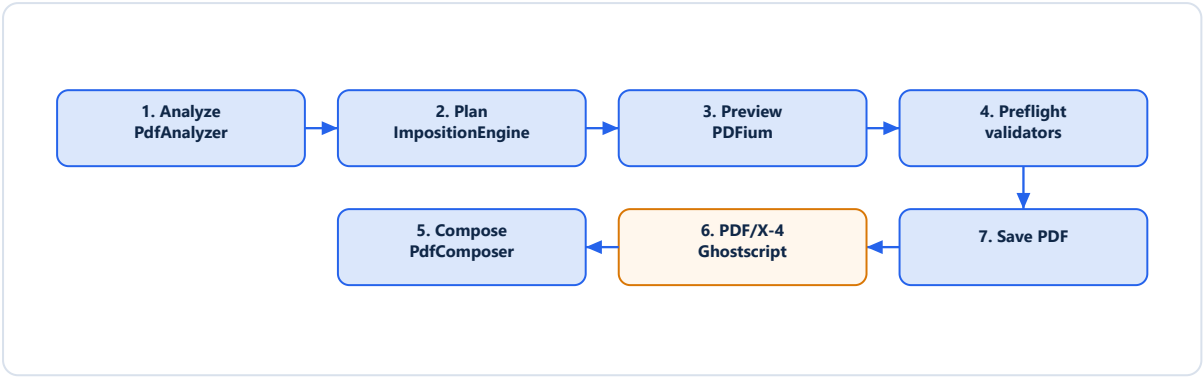


Fig. Processing pipeline: from source file to finished PDF (step 6 optional).

1. **Load & analyze** — `PdfAnalyzer` reads page count, dimensions, boxes, fonts, images, transparency.
2. **Plan** — `ImpositionEngine.Plan()` picks the strategy from `Template.Binding` and populates `ImpositionJob.SheetLayouts` (sheets with page positions — `PagePlacement`).
3. **Preview** — `PdfPreviewRenderer` (PDFium) rasterizes sheets; `ImpositionCanvas` draws them with LOD heuristics (detail level driven by zoom).
4. **Preflight** — `PrepressValidationEngine` runs all validators and produces a `PrepressValidationReport`.
5. **Compose** — `PdfComposer` (PdfSharpCore) places pages on sheets, draws marks and overlay.
6. **Convert (optional)** — `ExportPipeline` invokes Ghostscript with a PDF/X definition file and ICC profile.
7. **Save** — the finished PDF at the output location.

> **Two geometry compositors:** sheet geometry exists in two places — the output pipeline (`PdfComposer` / PdfSharpCore) and the preview (`ImpositionCanvas` / PDFium). The shared seam for fitting bleed into a cell is the BleedFit logic; output is bleed-aware (TrimBox → cell).

3. Installation & requirements

3.1. System requirements

Item	Requirement
Operating system	Windows 10 / 11 (64-bit)
Architecture	x64
.NET	No separate install — the release is self-contained
Ghostscript	Bundled (`Assets/ghostscript` , version 10.07) — for PDF/X-4 conversion
Memory	4 GB RAM minimum recommended (multi-sheet PDF/X-4 export is intensive)
Disk	~300–500 MB (with bundled Ghostscript and ICC profiles)

3.2. Installation

The application ships as an installer (Inno Setup) generated by the `build/pack.ps1` script. After installation, launch it from the Start menu or directly via `imPRESS Studio.exe`.

Dynamic files (configuration, templates, licenses, logs, the hot folder database) are **never** written into the Program Files directory — they go to `%APPDATA%` (roaming data) and `%LOCALAPPDATA%` (logs, SQLite, cache). See chapter 17. Files & data locations.

3.3. First run

1. Launch the app — on first start the data directories are created and the default configuration is loaded.
2. If no active license is found, the activation window appears (see chapter 4).
3. Built-in templates and marks profiles are seeded on first run.

4. Licensing & activation

imPRESS Studio uses a machine-bound license. The machine identifier is a hash computed from the computer's hardware parameters (`MachineFingerprint` , using the `System.Management` module).

4.1. Activation process

1. Open the *License* window (in the header) or wait for the activation window at startup.
2. Copy the **machine identifier** (fingerprint) — a unique hash of your computer.
3. Send it to your vendor when ordering a license.
4. You will receive an **activation key** (a Base64 string) or a **license file** (`.json`).
5. Paste the key and click *Activate license* — or load the license file.

4.2. License editions

Edition	Description
Trial	30 days, full functionality
Standard	Basic production edition
Professional	Full edition with PDF/X-4 and CLI support

4.3. License gate in the CLI

CLI export is subject to the same license gate as the GUI. If the license is invalid, the `impose` command returns exit code `2` and prints an activation instruction.

> **Note:** the license is tied to a specific computer. Changing the motherboard, CPU or reinstalling the OS may require re-activation. The license file lives in `%APPDATA%\imPRESS Studio\licenses` — copy it before reinstalling.

5. User interface

The main window is divided into three columns plus a header and a status bar.

Area	Contents
Header	Logo and buttons: *Template*, *Export options*, *Help*, *License*, *About*, plus the *Plan* and *Export* actions.
Left panel	Tabs: *Template* (current template), *Source PDF files* (loaded files), *Output file*, *Actions*, *Document* (page, sheet, signature counts).
Centre panel	Interactive sheet preview with a toolbar (zoom, navigation, options) and a thumbnail strip.
Right panel	Job statistics: sheets, signatures, paper usage, cost estimation, binding info.
Status bar	Current operation status and preview usage hints.

5.1. Quick start — five steps

1. **Pick or create a template** — the *Template* button in the header or *Manage templates* in the left panel.
2. **Load a PDF** — + *Add files…* (`Ctrl+O`) or drag a file onto the window. Selecting several files and clicking *Load selected* merges them into one document.
3. **Plan the imposition** — `Ctrl+P` or *Plan*.
4. **Check the preview** — navigate with ← →, zoom with the wheel, toggle marks, show front/back or spread.
5. **Export** — `Ctrl+E`, choose the range, PDF standard and optionally PDF/X-4.

> **Tip:** test your first run on a short file (e.g. 16 pages) with a simple "A5 — saddle, 2×1" template to understand how pages are arranged on sheets.

6. Imposition templates — full reference

A template (`Template`) is a complete imposition work plan stored as JSON. Below is the full field reference of the model.

6.1. Top-level fields

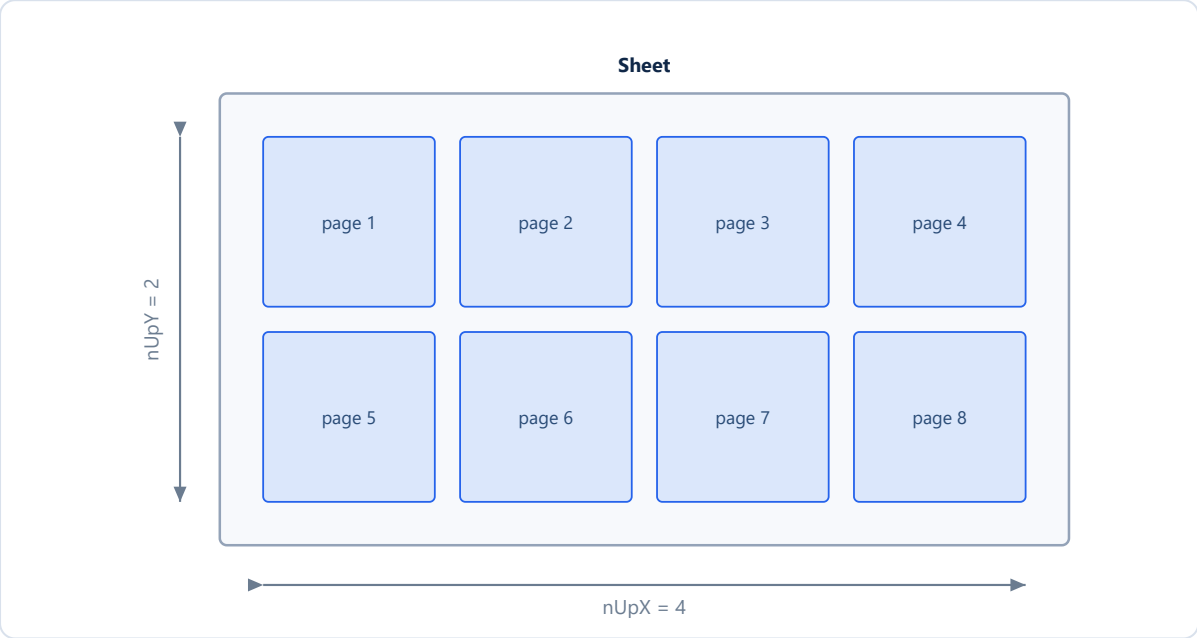


Fig. N-up layout: nUpX × nUpY pages per sheet side.

Field (JSON)	Type	Default	Description
`id`	string	auto	Template identifier (unique in the library; auto-generated when empty).
`name`	string	—	Display name.
`binding`	BindingType	—	Binding type (see chapter 7).
`pagesPerSignature`	int	8	Pages per signature (saddle/perfect; must be a multiple of 4).
`nUpX` / `nUpY`	int	2 / 1	Pages across / down (N-up).
`sheet`	SheetSize	—	Sheet dimensions (`widthMm` , `heightMm` , `name`).
`trim`	TrimSize	—	Final page size after trimming (`widthMm` , `heightMm`).
`margins`	LayoutMargins	—	Margins and spacing (see 6.2).
`marks`	MarksOptions	—	Marks (legacy toggles; see chapter 8).
`marksProfileRef`	string?	null	Reference to a marks profile; when set, takes precedence over `marks` .
`creep`	CreepOptions	—	Creep compensation (see chapter 9).
`saddle`	SaddleOptions	—	Saddle-stitch sheet layout.
`fold`	FoldOptions	—	Fold options (panel count).
`gatefold`	GatefoldOptions	—	Gatefold options (style + reduction).
`roll`	RollOptions	—	Roll-fed options.
`stepAndRepeat`	StepAndRepeatOptions	—	Step & repeat options.
`overlay`	OverlayOptions	—	Logo/image overlay.
`duplex`	bool	true	Double-sided printing (work-and-back).

6.2. Margins (`LayoutMargins`)

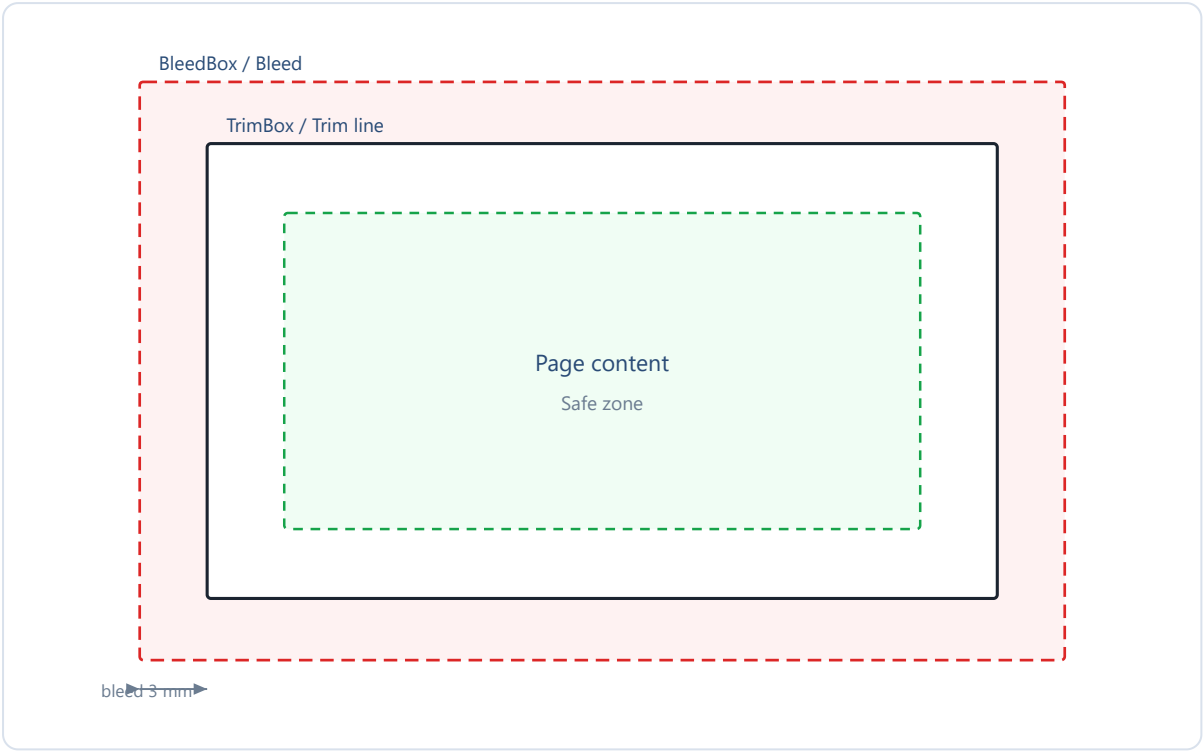


Fig. Page box model: BleedBox ⊃ TrimBox ⊃ safe zone.

Field	Default	Description
`bleedMm`	3.0	Bleed — artwork extending past the trim lines.
`safeZoneMm`	5.0	Safe zone (inset from trim).
`gutterMm`	0.0	Gutter — gap between adjacent pages.
`spineMm`	0.0	Spine width for perfect bound.
`sheetMarginMm`	10.0	Sheet edge margin (press gripper).

6.3. Template validation

`Template.Validate()` enforces correctness rules before planning:

- Sheet and trim dimensions must be positive; N-up X/Y positive; margins non-negative.
- For saddle and perfect binding: pages per signature positive and a multiple of 4.
- 8-page signature (saddle): page count a multiple of 8.
- Stacked copies mode: at least 2 copies per sheet.
- Mode-specific sheet-fit check (e.g. saddle verifies required width/height = 2×trim + gutter + 2×margin).

6.4. Managing templates

- Pick a ready-made template from the library.
- Create a new template from scratch (the *New* button).

- Edit parameters and save under your own name (a JSON file).
- Import/export templates as `.json` files — to share between workstations.

> **Template presets (1.4.5):** presets automatically swap ISO ↔ US sets when toggling mm/inch, and the page range resets when the source file changes.

7. Binding types & layout strategies

Each binding type has a dedicated strategy (`IBindingStrategy`) chosen by `ImpositionEngine` based on the template's `binding` field.

7.1. Saddle stitch (SaddleStitch)

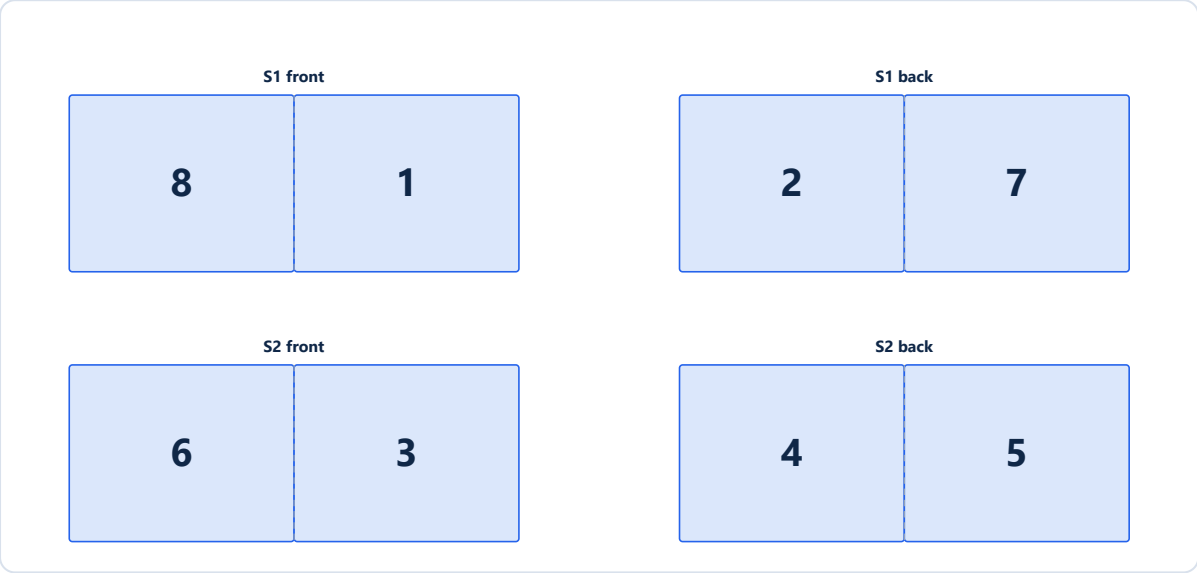


Fig. Page order in saddle stitch (8 pages, 2 sheets).

Folded signatures stapled at the spine. Typically 8 or 16 pages per signature. Ideal for thin publications (up to ~60 pages). Requires creep compensation at higher page counts. Layout modes (`SaddleSheetLayout`):

Mode	Description
`TwoUp`	Classic 2-up: one folded spread per sheet (4 pages per sheet).
`StackedCopies`	N identical copies of the spread stacked in rows on a larger sheet. The sheet is cut into strips first, then folded — a typical digital workflow.
`EightPageSignature`	Classic 8-page form: a 2×2 grid with the top row rotated 180° (heads together), folded twice without cutting. Page count must be a multiple of 8.

7.2. Perfect bound (PerfectBound)

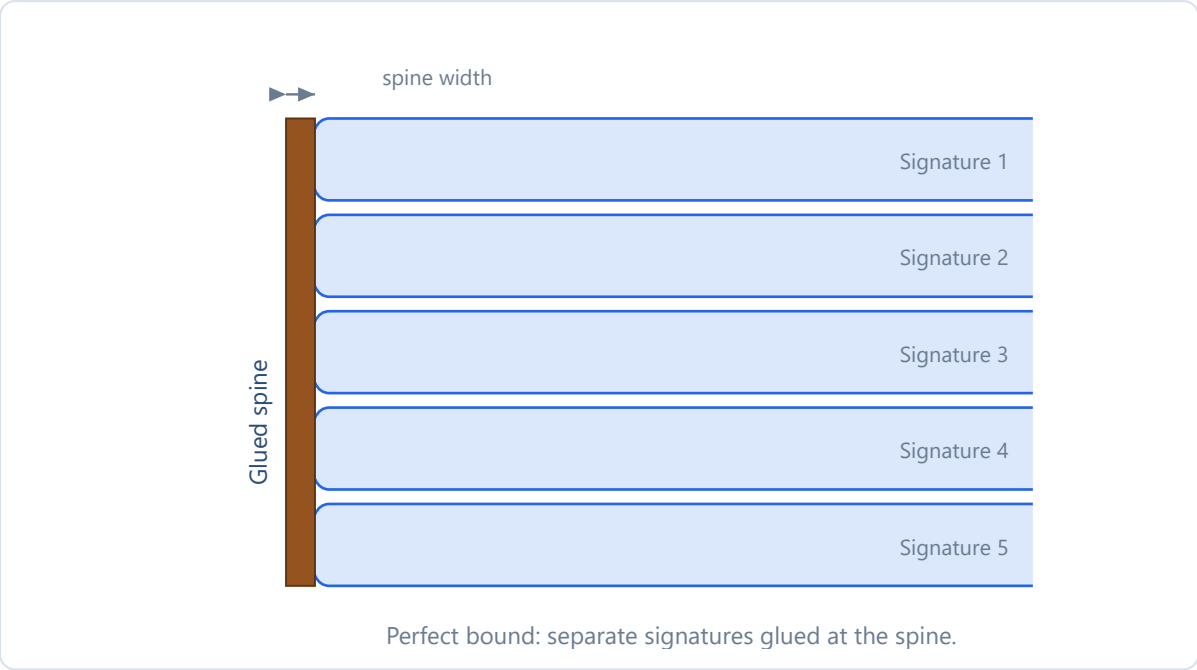


Fig. Perfect binding: stack of signatures and the spine.

Each signature is a separate section glued at the spine. Used for thicker books (over ~50 pages). Requires a spine margin (`spineMm`) whose width depends on the total book thickness.

7.3. Wire / spiral (Wiro)

Wire or spiral binding — single pages with punched holes.

7.4. N-up ganging (Ganging)

Multiple different flyers/pages on one sheet, no binding. The strategy cycles through successive source pages and never rotates copies.

7.5. Folding: Z-fold, Gatefold, Accordion

Fig. Folding schemes: Z-fold, 4-panel gatefold, accordion (fold lines).

The sheet is folded without cutting (leaflets, brochures, maps). Panel count is in `FoldOptions.PanelCount`:

> **Geometry change in 1.5.11.** Up to 1.5.10 the panel width was derived by *dividing the sheet*: `(sheet − margins − gutters) ÷ panels`, and the trim size was only a rectangle the artwork was fitted into. The fold lines therefore moved with the stock — a DL leaflet with 99 mm panels imposed on SRA3 came out with its panels 143.33 mm apart, so a folder creasing at 99 mm creased through the artwork. Since 1.5.11 **the panel width comes from the product**: panels are laid out at the entered size, touching at the fold lines, with the whole block centred on the sheet and the leftover paper left blank. The sheet is stock, not a design parameter.

All of this geometry is computed in one place — `FoldGeometry` — shared by the layout strategy, template validation, the preview and the marks renderer, so all four agree on where the sheet gets creased:

Method	Returns
`PanelsFor(template)`	Panel count; gatefold derives it from its style (3 wings or 4 panels), the others from `FoldOptions.PanelCount`.
`PanelSizeMm(template)`	The size of *one* panel — i.e. the page size the source PDF must carry.
`FlatSizeMm(template)`	The unfolded product size (panel × panel count).
`Compute(template)`	A `FoldLayout` in points: panel width and height, block origin, gutter, fold-line positions (`FoldLinesPt()`) and the `FitsSheet` flag.

Size interpretation mode (`FoldOptions.SizeMode`) 1.5.11

A folded product can be described either way round; the choice is stored in the template:

Value	Meaning of `trim`	Panel width
`Panel` (default)	One panel — the page size of the incoming PDF.	`trim.widthMm`
`FlatProduct`	The unfolded product, e.g. "A4 landscape folded to DL".	`trim.widthMm ÷ panel count`

Templates written before 1.5.11 deserialize as `Panel` — exactly what their trim already meant — so their sheets are unchanged.

Panel fit check 1.5.11

Up to 1.5.10 only gatefold had a fit check; Z-fold and accordion fell through to the generic N-up test, which — with the `NUpX = 1` every fold preset ships — asked whether *one* panel fits. Three panels of 210 mm on a 297 mm sheet passed validation, and the strategy then produced panels overlapping by 114 mm. All three folding bindings now share one check:

```
requiredWidth  = panels × panelWidth + (panels − 1) × gutter + 2 × sheetMargin
requiredHeight = panelHeight + 2 × sheetMargin
```

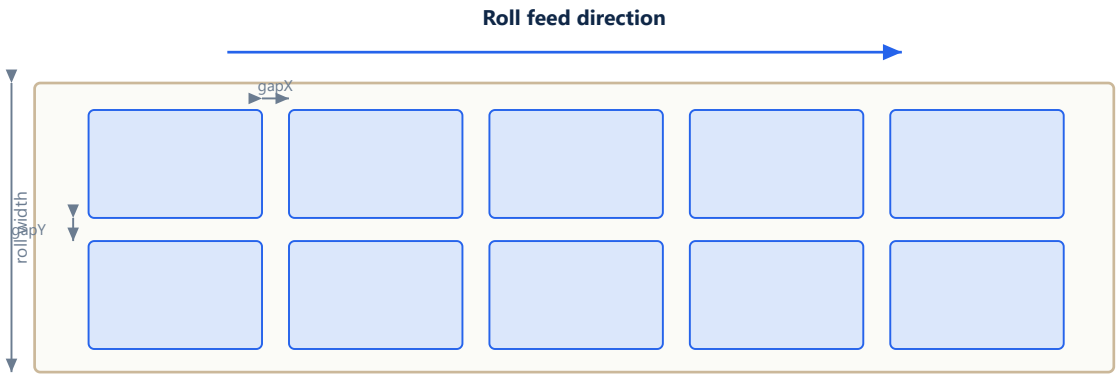
The message names what it measured: "*3 panels of 210.0mm*" or "*unfolded product 297×210mm = 3×99.0mm*".

Fold marks 1.5.11

Up to 1.5.10 fold marks were drawn for saddle stitch only, even though every fold preset ships with them enabled. Since 1.5.11 dashed ticks are drawn at the top and bottom sheet edge at the positions returned by `FoldLayout.FoldLinesPt()` — identically in the preview and in the exported PDF.

Type	Panels	Description
Z-fold	3	Three panels folded in a Z shape.
Gatefold	3 or 4	`ThreePanel` (6 pages, two wings fold inward) or `FourPanel` (8 pages, classic gate). Wings are narrowed by `foldReductionMm` (default 2 mm) so they don't collide at the centre line.
Accordion	4+	Multiple panels folded alternately (concertina).

7.6. Roll-fed step-and-repeat (RollFed) 1.2



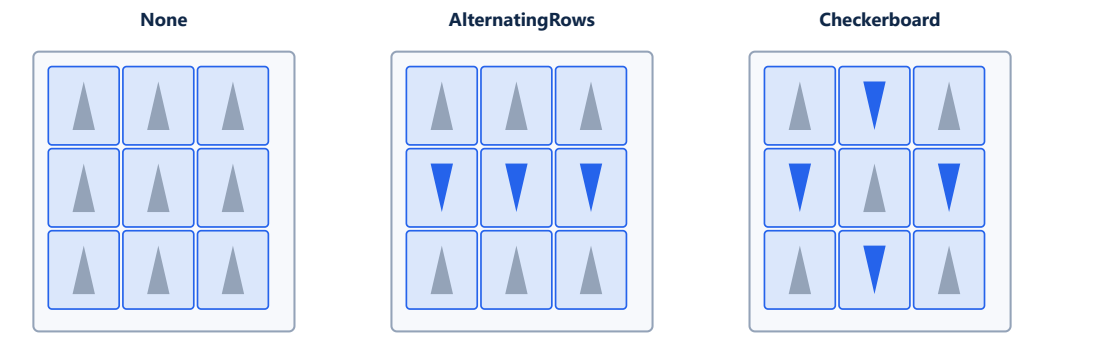
Roll-fed: variable repeat + gapX/gapY spacing; optional basic nesting (90° rotation).

Fig. Roll-fed step-and-repeat on a continuous roll.

Sticker/label production on a continuous roll (web-press). Sheet width = physical roll width. Options (`RollOptions`):

Field	Default	Description
`maxRollLengthMm`	0 (unbounded)	Maximum length of one output segment; 0 = a single segment of computed length.
`gapXMm` / `gapYMm`	2.0	Horizontal/vertical gap between items.
`repeatCount`	1	Total number of items to produce (variable repeat).
`allowRotation`	false	Evaluate 0°/90° rotation and pick whichever fits more steps (basic nesting).

7.7. Sheet-fed step & repeat (StepAndRepeat) 1.2



Step & Repeat: 180° rotation patterns (triangle = copy orientation) for better nesting.

Fig. Step & Repeat: copy rotation patterns (None / AlternatingRows / Checkerboard).

Replicates ONE source page into an NxM grid with optional rotation patterns and pitch (centre-to-centre) spacing. Differs from Ganging in that Ganging cycles through multiple pages and never rotates copies.

Field	Description
`rotationPattern`	`None`, `AlternatingRows`, `AlternatingColumns`, `Checkerboard` — 180° rotation patterns for better nesting of irregular shapes.
`spacingMode`	`GapBetween` (like Ganging) or `CenterToCenter` (machine pitch; gutter ignored, grid centred).
`horizontalPitchMm` / `verticalPitchMm`	Centre-to-centre distance (CenterToCenter mode only).

8. Margins, bleed & printer marks

8.1. Print concepts

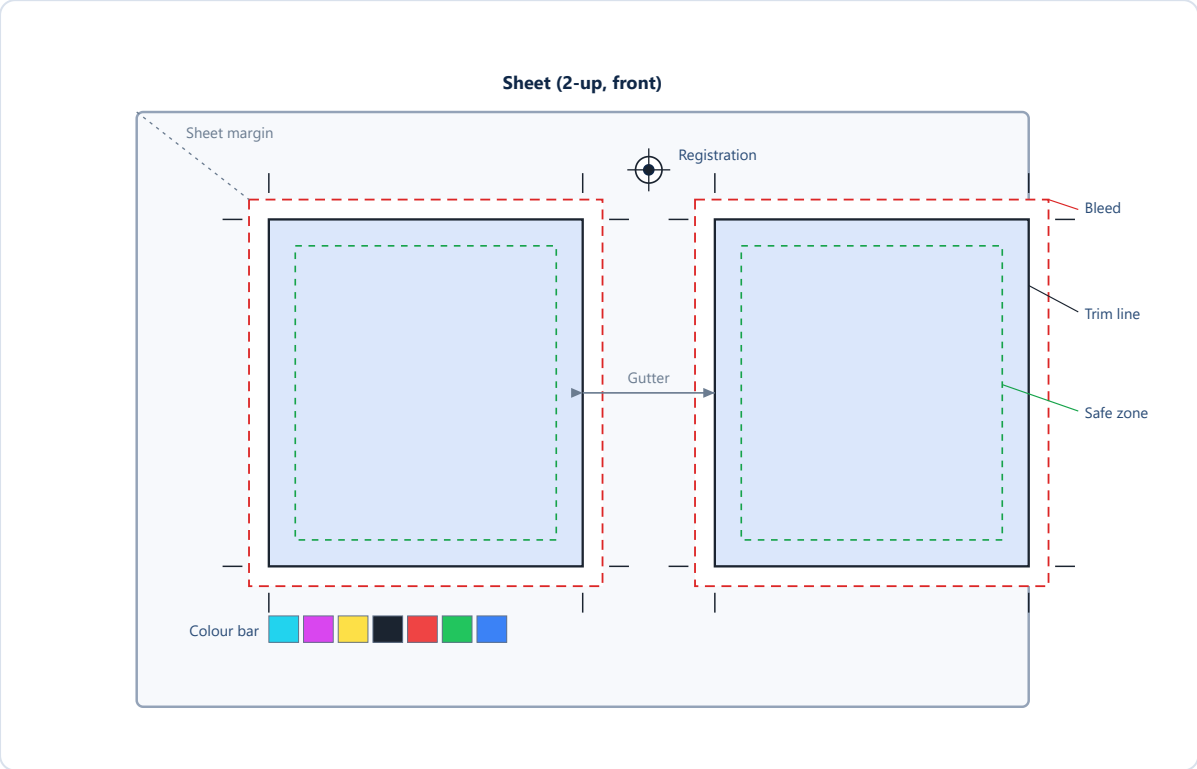


Fig. Anatomy of a 2-up sheet: bleed, trim, safe zone, gutter, margin, registration, colour bars, crop marks.

Bleed

Artwork extending past the trim lines — typically 3 mm. Prevents white edges after a slightly inaccurate cut.

Safe zone

Minimum distance of text from the trim line — typically 5 mm.

Gutter

Gap between adjacent pages on the sheet.

Sheet margin

Distance from the sheet edge to the first page — required by the press gripper.

Spine

Spine width for perfect bound (paper thickness × page count).

8.2. Printer marks (`MarksOptions`)

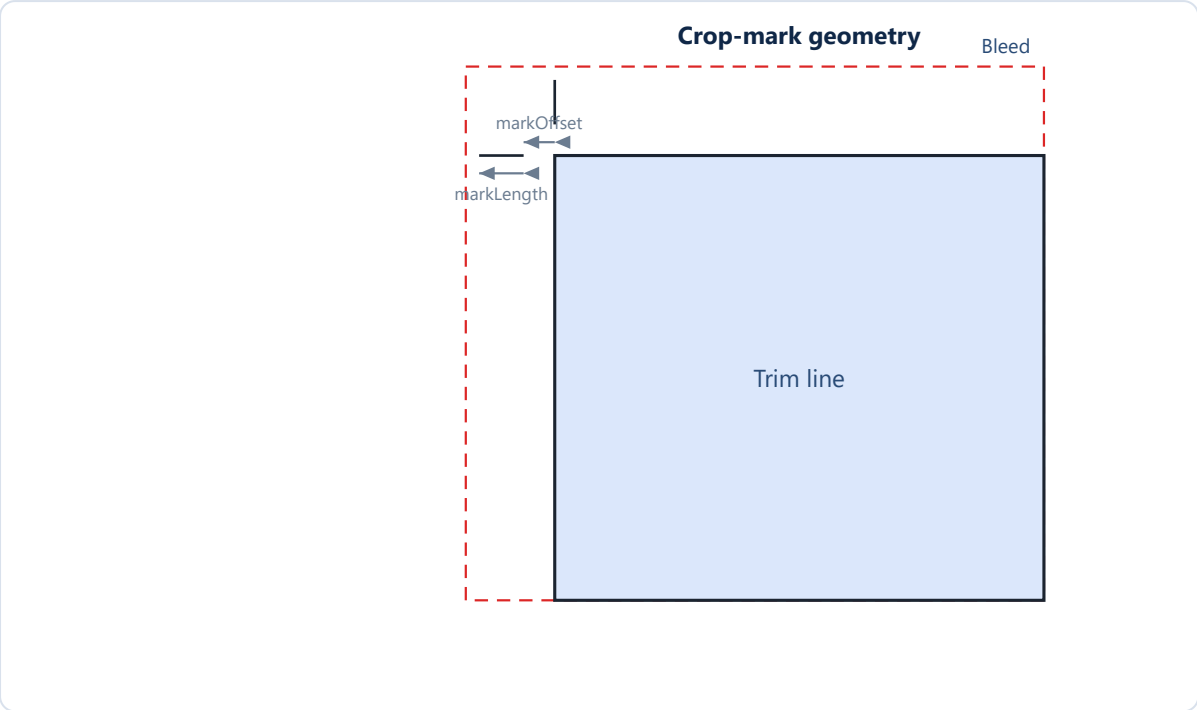


Fig. Crop-mark geometry: markLength and markOffset.

Field	Default	Description
`crop`	true	Crop marks at trim corners.
`registration`	true	Registration marks (crosshairs centring CMYK separations).
`colorBars`	true	CMYK control bars along an edge.
`foldMarks`	true	Fold marks (creasing).
`markLengthMm`	5.0	Crop mark length.
`markOffsetMm`	3.0	Offset of marks from the trim edge.
`cutterMarks`	None	Contour markers: `SummaOPOS` (4 black squares at corners) or `SummaOPOSXY` (OPOS + a job-ID label).
`cutterMarkSizeMm`	3.0	Marker square edge (Summa default 3 mm).
`cutterMarkOffsetMm`	5.0	Clearance between the cut area and the marker (Summa requires ~5–8 mm).

8.3. Marks profiles 1.3

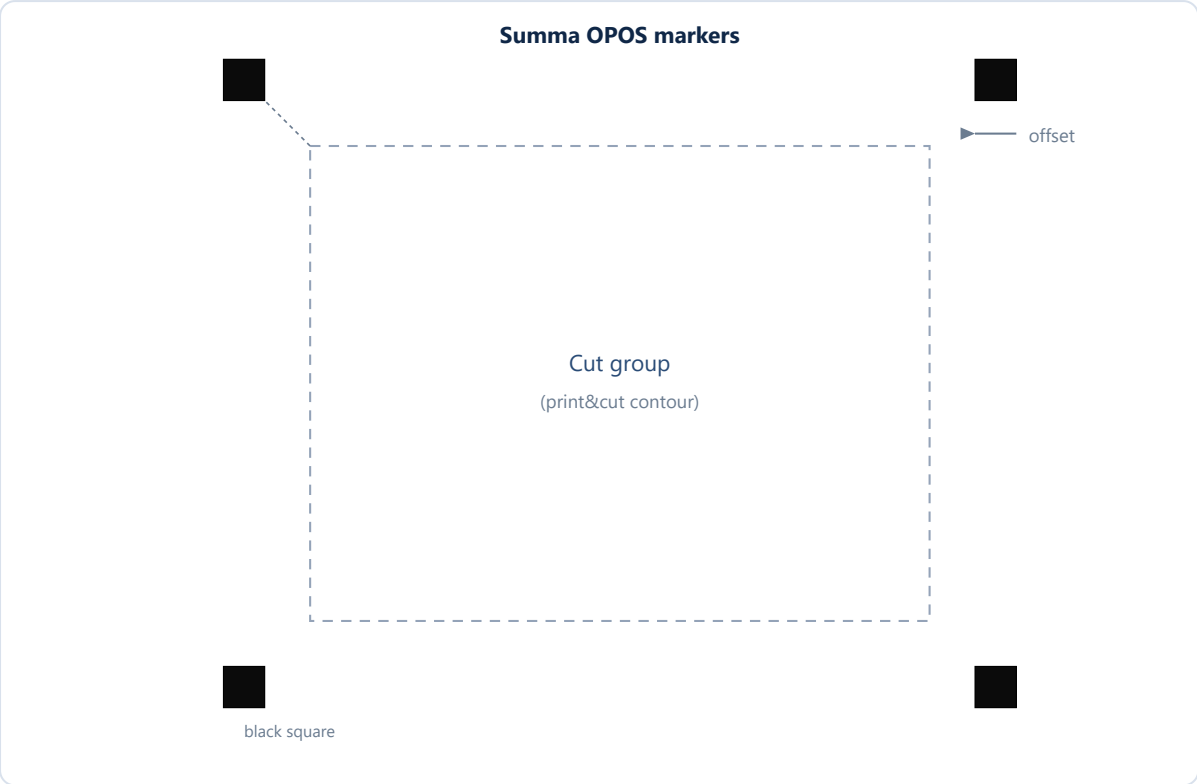


Fig. Summa OPOS contour markers around the cut area.

Newer templates use **marks profiles** referenced via `marksProfileRef` instead of flat toggles. Profiles are stored in a file-backed store and seeded with built-ins on first run. The resolver uses the profile when the reference is non-empty, otherwise it falls back to the legacy `marks` toggles. Profiles are edited in a dedicated editor (MarksProfileEditor) and support, among others: registration mark style and position, colour-bar style, mark ink colours, barcodes, and the K grayscale wedge (1.4.4).

8.4. Overprint simulation

The preview can simulate overprint using multiply blend — it shows the darkening when CMYK inks overprint, helping you anticipate the press result.

9. Creep compensation

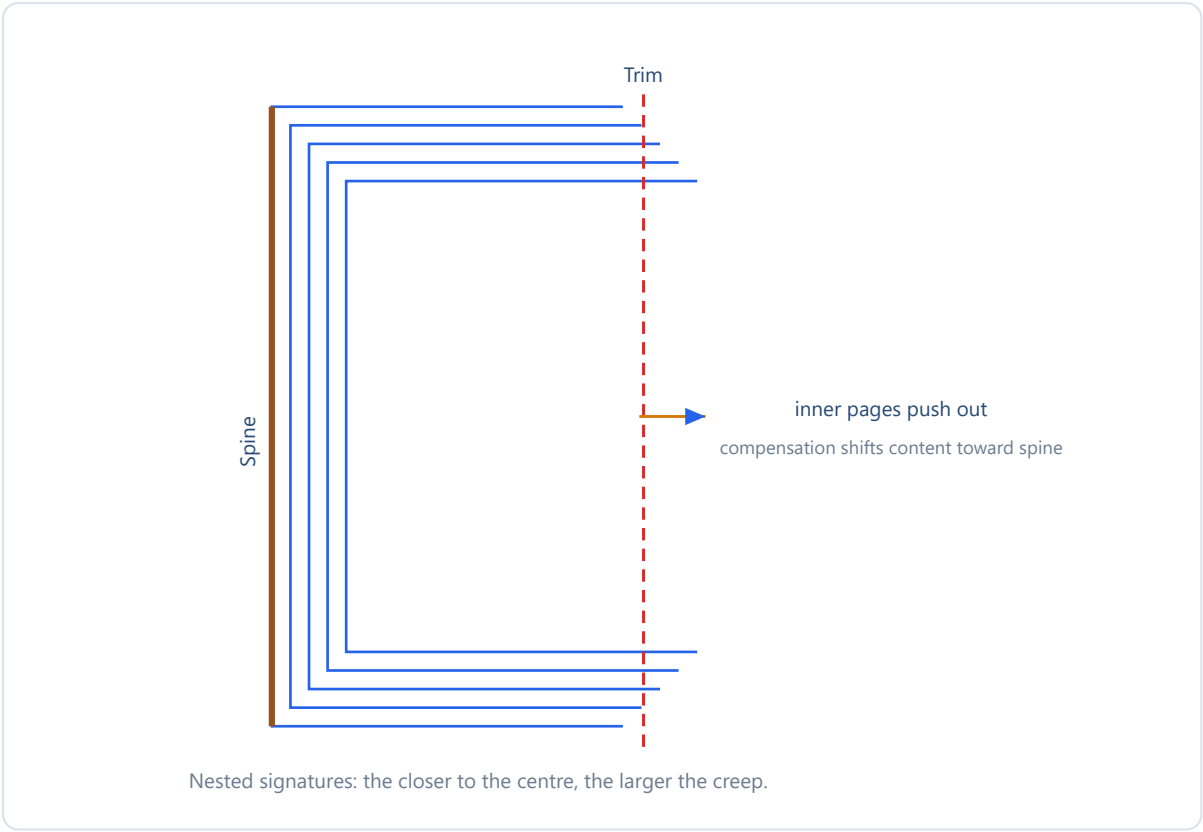


Fig. Creep and its compensation in saddle stitch.

In saddle stitch all signatures nest inside each other. The inner pages "push out" past the outer edge by the thickness of the preceding sheets. After trimming the outer edge, the inner pages' margins become narrower — by several millimetres on thicker publications.

Creep compensation shifts the inner pages' content toward the spine so that, after trimming, the margins are identical on all pages.

9.1. Options (`CreepOptions`)

Field	Default	Description
`enabled`	false	Enable compensation.
`paperThicknessMm`	0.1	Single-sheet thickness (e.g. 0.1 mm for 100 gsm; 0.13 mm for 115 gsm).

9.2. How to enable

1. Open the template manager, the *Creep* tab.
2. Tick *Enable compensation* and enter the paper thickness.
3. Turn on the *Creep* option in the preview — orange dashed lines show the compensation per signature.

Note: creep compensation applies only to saddle stitch. With perfect binding each section is separate and the problem does not arise.

10. Source PDF files

The *Source PDF files* tab in the left panel manages the list of input documents.

- **+ Add files...** (`Ctrl+O`) — a dialog to select multiple PDFs at once.
- **Drag and drop** — drop PDF files onto the window (Win32 drag&drop, hence the STA requirement).
- **Checkboxes** — select files to load; more than one + *Load selected* merges them into one document in list order.
- **Arrows ▲▼** — reorder before merging.
- **Info** — details of the selected PDF: page count, dimensions, trim, metadata, fonts, transparency, encryption, tagging, OCG (layers).
- ✕ — remove the selected file; **Remove unchecked** — clear files without a checkbox.

> **Tip:** the *Info* panel is invaluable for diagnostics — it reveals PDF encryption, mixed page sizes, missing bleed or non-embedded fonts.

11. Sheet preview

11.1. Navigation

- **Mouse wheel** — zoom relative to the cursor.
- **Drag** — pan the view.
- **←** / **→** — previous / next sheet.
- **Thumbnails** — jump to a sheet; the **"Sheet X / Y"** field is the counter.
- `Ctrl+0` or `F` — fit to window; `R` — rotate 90°; `Ctrl++` / `Ctrl+-` — zoom.
- Quick-zoom buttons: *50%*, *Fit*, *200%*.

11.2. What you can visualize

- **Back / Front** (⇄) — toggle sheet sides (double-sided printing).
- **Spread** — front and back side by side, to check registration.
- **Marks** — crop, registration, colour bars, fold.
- **Creep** — orange compensation lines per signature.
- **Overprint** — multiply-blend simulation.
- **View rotation** (↻) — 90° to check orientation on press.

> **Performance (LOD):** the `ImpositionCanvas` control selects a rendering detail level based on the current zoom, keeping it smooth with many sheets.

12. Preflight — prepress validation

Preflight is an automated check of PDF correctness before printing. The deep analyzer (`PdfPreflightAnalyzer`) inspects the file, and the validation engine (`PrepressValidationEngine`) runs a set of validators producing a report (`PrepressValidationReport`) with severity levels.

12.1. Validators 1.1.1

Validator	What it checks
BleedValidator	Presence and amount of bleed.
PageBoxConsistencyValidator	Consistency of page boxes (MediaBox/TrimBox/BleedBox).
PageSizeValidator	Page dimensions (mixed / unexpected sizes).
ImageDpiValidator	Image resolution (DPI too low for print).
RgbColorSpaceValidator	RGB detected where CMYK is expected.
MixedColorSpaceValidator	Mixed colour spaces in the document.
FontEmbeddingValidator	Non-embedded fonts.
TransparencyValidator	Transparency (risk with older RIPs).
SpotColorValidator	Spot colours — detection and count.
PdfStandardValidator	Compliance with the declared PDF/X standard.

12.2. Strict mode and the report

In strict preflight mode, warnings halt the export. The preflight report can be reviewed in a dedicated dialog (`PreflightReportDialog`), and a summary is produced by `PreflightSummaryGenerator` . In the export pipeline, preflight runs before composition, with an optional confirmation callback (`confirmAfterPreflight`).

13. Export & PDF/X-4 conversion

> **Since 1.5.12 all three standards are packaged by the native engine.** Up to 1.5.11 the imPRESS Export Engine handled PDF/X-4 only, and X-1a and X-3 required Ghostscript. Ghostscript is now entirely *optional* — and it is not distributed with the application, being separately licensed AGPL software.

The three flavours differ in exactly three ways, handled in one place (`ImpressExportEngine.PackageAsPdfX`):

Standard	PDF version	Colour	Transparency
PDF/X-1a:2001 (ISO 15930-4)	1.4	DeviceCMYK, DeviceGray and spot only	forbidden
PDF/X-3:2003 (ISO 15930-6)	1.4	device-independent (ICCBased) permitted	forbidden
PDF/X-4:2010 (ISO 15930-7)	1.6	as X-3	permitted

How X-1a and X-3 handle transparency 1.5.12

Both are defined on PDF 1.4, which has no transparency. The engine does **not flatten**: turning genuinely non-opaque artwork into opaque marks means compositing the page, which is a rasteriser's job and would silently convert vector text into pixels. Instead `NativeTransparencyFlattener` separates two cases:

Construct	Treatment
`/Group << /S /Transparency >>` with no other transparency present	inert — removed
`/CA 1` , `/ca 1` , `/BM /Normal` , `/SMask /None`	inert — removed
`/ca` or `/CA` < 1	paints — export refused
soft mask (`/SMask`) in graphics state or in an image	paints — export refused
blend mode other than Normal / Compatible	paints — export refused

Inert entries are stripped **only when the whole document** is free of transparency that paints: an isolated or knockout group becomes observable once real transparency is in play.

Why the removal happens on the finished file 1.5.12

PdfSharpCore stamps `/Group << /CS /DeviceRGB /S /Transparency >>` onto *every* page it serialises — measured on a blank page with no content and no transparency. Removing the key through the object model achieves nothing, because the writer puts it back. The groups are therefore taken out of the written file, and the replacement is byte-for-byte the same length (spaces are legal whitespace between PDF tokens), which keeps the cross-reference table valid without a rebuild.

After saving, the file is **re-read and verified**: surviving transparency (and, for X-1a, surviving DeviceRGB) fails the export. A library change surfaces as a refused export rather than as a file claiming conformance it does not have.

13.1. The export process

After pressing *Export* (`Ctrl+E`), `ExportPipeline.RunAsync` performs:

1. **Source analysis** — pages, fonts, images, transparency.
2. **Preflight** — correctness check (optionally strict).
3. **Composition** — placing pages on sheets + marks + overlay.
4. **Optional PDF/X-4** — natively via the imPRESS Export Engine (with vector-RGB conversion through ICC) or via Ghostscript.
5. **Save** — the finished PDF at the output location.

13.2. Export options (`ExportOptions`)

Option	Description
PDF/X-4	ISO 15930-7 standard. Supports transparency, layers and ICC profiles.
ICC profile	Output colour profile (e.g. ISOcoated_v2_eci for offset on coated stock).
Strict preflight	Halt the export on warnings.
Sheet range	E.g. `1-5` , `2,4,6` , `1,3-7` (the `RangeExpression` parser).
Page numbering	Numbering stamp (`PageNumberingOptions`).
Cutting guide	A page with a guillotine cut schematic + positions table (`--cutting-guide` in the CLI).

13.3. PDF/X-4 — requirements and profiles

PDF/X-4 (ISO 15930-7) is the print standard supporting transparency, layers and ICC profiles, recommended by most print shops. Since 1.5.2 the packaging is done by default by the native **imPRESS Export Engine** (OutputIntent + XMP + TrimBox + PDF 1.6, vector RGB→CMYK conversion through the ICC profile using the Windows colour engine); files with RGB images and the X-1a/X-3 modes are handled by Ghostscript (bundled, version 10.07) with a definition file (`PdfXDefinitionFile` / `PDFX_def.ps`) and an ICC profile.

Profile	Use
ISOcoated_v2_eci / PSOcoated_v3	Offset, glossy coated stock.
PSO_Uncoated_ISO12647 / PSOuncoated_v3_FOGRA52	Offset, uncoated stock.
eciRGB_v2 / sRGB	Digital print, RGB output.

> **Ghostscript and SAFER:** Ghostscript's SAFER mode requires passing `--permit-file-read` for the ICC profile in `PDFX_def.ps` . Real Ghostscript errors are emitted on stdout (not only stderr).

14. Job statistics

After planning the imposition, the right panel shows statistics computed by `JobStatistics`:

Metric	Description
Sheets and signatures	Number of physical sheets off the press.
Filled / empty slots	How many sheet positions are occupied by pages.
Paper area	Gross and net (after trimming) in m^2.
Estimated weight	Based on grammage (gsm).
Utilization / waste	Percentage of area reaching the publication vs. trimmed away.
Estimated cost	From the "Price per sheet" field.
Binding info	Type, pages per signature, bleed, gutter.

> **Tip:** statistics help compare template variants — e.g. 2×1 on SRA3 vs. 2×2 on B2 — to choose the more cost-effective one.

15. Hot Folder — automation 1.1.0

The hot folder subsystem monitors designated directories and automatically processes files dropped into them. The architecture is based on `Microsoft.Extensions.Hosting` and is fault-tolerant (retry/backoff via Polly, idempotency via a SQLite ledger).

15.1. Components

Component	Role
HotFolderConfig / Store	Folder configuration (`hotfolders.json`), validation.
Watching	Watcher (FileSystemWatcher or polling), event debouncer, file-stability probe (waits until the file stops growing).
Queueing	Job queue, backoff policy on errors.
State	SQLite ledger (`SqliteProcessingLedger`) — tracks processed files, prevents reprocessing.
Processing	Ghostscript throttle (concurrent conversion limit), output naming strategy, file transitions.
Hosting	Host and manager (start/stop/reload), status snapshot, diagnostic metrics.

15.2. Management

Hot folders are configured in the manager dialog (`HotFolderManagerDialog`) and the editor (`HotFolderEditDialog`), or from the CLI (see chapter 16). The default concurrent Ghostscript limit is 2 (configurable in global settings).

> **Data location:** the configuration and ledger database live under `%APPDATA%` / `%LOCALAPPDATA%` (never in Program Files) — see chapter 17.

16. Command-line interface (CLI)

imPRESS Studio recognizes command-line arguments (System.CommandLine parser). With no arguments it launches the GUI; with a verb argument it runs headless.

16.1. The `impose` command

```
"imPRESS Studio.exe" impose \
    --source input.pdf \
    --output output.pdf \
    --template template.json \
    --pdfx4
```

Option	Required	Description
`--source`	yes	Path to the source PDF.
`--output`	yes	Output file path.
`--template`	yes	Template JSON file.
`--pdfx4`	no	Convert to PDF/X-4 via Ghostscript.
`--gs`	no	Path to Ghostscript (if not on PATH).
`--cutting-guide`	no	Append a cutting-guide page.

Exit codes: `0` — success; `1` — error; `2` — missing/invalid license.

16.2. The `license` command

```
"imPRESS Studio.exe" license fingerprint
"imPRESS Studio.exe" license info
"imPRESS Studio.exe" license activate --key BASE64_KEY
```

16.3. The `hotfolder` command

```
"imPRESS Studio.exe" hotfolder run      # foreground daemon (Ctrl+C stops)
"imPRESS Studio.exe" hotfolder list     # status of all folders
"imPRESS Studio.exe" hotfolder start <GUID>
"imPRESS Studio.exe" hotfolder stop  <GUID>
"imPRESS Studio.exe" hotfolder reload   # reload hotfolders.json
```

The `hotfolder run` command must be the process's first command — it is intercepted in `Program.Main` before the parser, because it requires the host to be wired up around the CLI.

17. Files & data locations

The `AppPaths` helper centralizes all paths. Roaming data goes to `%APPDATA%\imPRESS Studio`, local data to `%LOCALAPPDATA%\imPRESS Studio`.

Location	Contents
`%APPDATA%\…\templates`	Imposition templates (JSON).
`%APPDATA%\…\presets`	Presets.
`%APPDATA%\…\licenses`	License file (`.json`).
`%APPDATA%\…\icc`	User ICC profiles.
`%APPDATA%\…\export-presets`	Export presets.
`%APPDATA%\…\localization`	Localization overrides (custom .json packs).
`%APPDATA%\…\marks-profiles`	Marks profiles.
`%LOCALAPPDATA%\…\logs`	Logs (Serilog).
`%LOCALAPPDATA%\…\state`	State, hot folder SQLite database.
`%LOCALAPPDATA%\…\cache`	Cache.

18. Localization & units

The interface is fully localized through `LocalizationService` and the `LocExtension` markup extension. Built-in languages include Polish (pl), English (en), German (de), Spanish (es), French (fr), Italian (it), Japanese (ja), Korean (ko), Russian (ru) and Turkish (tr). Custom language packs (.json) can be added in the localization overrides directory.

Display units (`DisplayUnitsService`) support millimetres and inches. The data model is canonical in millimetres; conversion happens at the XAML boundary via `MmConverter`. When toggling mm/inch, template presets swap ISO ↔ US sets (1.4.5).

19. Keyboard shortcuts

Shortcut	Action
`Ctrl+O`	Load a PDF file
`Ctrl+P`	Plan the imposition
`Ctrl+E`	Export to PDF
`Ctrl++` / `Ctrl+-`	Zoom in / out
`Ctrl+0` / `F`	Fit to window
`R`	Rotate view 90°
`←` / `→`	Previous / next sheet
`F1`	In-app help
Mouse wheel	Zoom relative to cursor
Drag	Pan the view
Drag & drop PDF	Load a file by dropping

20. Troubleshooting

"No active license"

Open *License* and paste the activation key or load the `.json` file.

PDF/X-4 export doesn't work

Ghostscript is bundled, but for a custom path point to `gswin64c.exe` in the options (or `--gs` in the CLI). Remember the ICC profile read permission under SAFER mode.

Preview shows black pages

Check whether the PDF is encrypted (*Info* in the left panel). Restricted PDFs may require removing protection.

Pages run "off" the sheet

Trim + bleed + gutter + sheet margins exceed the sheet size. Pick a larger sheet or reduce margins/bleed. Template validation reports the required width/height.

"The file is in use by another process"

Close the output file in Acrobat/a browser, or choose a different path.

Slow export of large files

Multi-sheet PDF/X-4 export is intensive. Consider exporting without PDF/X-4 and converting in bulk via a script, or use hot folders with throttling.

21. Glossary

Imposition

Arranging pages on press sheets so that, after printing, folding and trimming, a publication with correct pagination results.

Signature

A group of pages printed on one sheet (typically 4, 8 or 16).

N-Up

Number of pages on one side of a sheet (e.g. 2×2 = 4 pages).

Trim

Final page size after trimming the bleed.

Bleed

Artwork extending past the trim lines.

Creep

Inner pages pushing out past the outer edge in saddle stitch.

PDF/X-4

The ISO 15930-7 print standard (transparency, layers, ICC).

Preflight

Automated correctness check of a file before printing.

Registration marks

Crosshair marks for aligning CMYK separations.

Pitch

Centre-to-centre distance between copies (step & repeat).

Nesting

Optimal nesting of shapes (rotations) for less waste.

OPOS

Summa plotter contour-marker system (print&cut).

OCG (Optional Content Groups)

PDF layers that can be toggled on/off.

Hot folder

A monitored directory — files dropped in are processed automatically.

LOD (Level of Detail)

Rendering detail level driven by zoom.

22. Version history (milestones)

Version	Key changes
1.0.x	First releases: imposition engine, saddle stitch and perfect bound, preview, PDF export, marks, creep compensation, CLI.
1.1.0	Hot Folder subsystem (queue, SQLite ledger, Ghostscript throttling, host).
1.1.1	Deep prepress preflight (PDF analyzer + validation engine with 10 validators).
1.2.x	Roll-fed step-and-repeat, sheet-fed step & repeat, Summa OPOS / OPOS-XY markers (print&cut), logo overlay (PNG/JPG/PDF on every sheet).
1.3.x	Marks profiles with an editor and a file-backed store; expanded mark options.
1.4.0–1.4.3	Further improvements to layouts, marks and the export flow.
1.4.4	Colour bar split off the parser; K grayscale wedge.
1.4.5	Template presets swap ISO ↔ US sets when toggling mm/inch; page range resets when the source file changes.
1.4.6–1.4.7	Stability and correctness fix pack (quarter-turn bleeds, duplex mirroring, Z-fold/Accordion ordering, resource leaks).
1.4.8	Ticket numbering, work-and-turn, front/back registration preview, sheet advisor, PDF/X-3, Preflight 2.0 (page content & annotations), CLI parity.
1.5.0	imPRESS Export Engine: native PDF/X-4 with vector RGB→CMYK conversion through ICC (WCS) — no Ghostscript; engine and PDF-version selection at export; native ink-coverage scanner; redesigned template manager (tabs, context-aware sections).
1.5.1	Fix for the silent template mutation (Run.Text TwoWay), localized preflight, messages naming the judged template.
1.5.2	Hot folders 2.0 (template picker, per-folder engine and PDF version, Advanced section, auto-Ghostscript); full page-box set (CropBox/BleedBox/TrimBox) on exported sheets.
1.5.4–1.5.6	Fixes and refinements to the export flow and preview.
1.5.7	The sheet strip moves into the inspector; clicking a thumbnail no longer discards the plan.
1.5.8	Template suggestions ranked by six weighted factors, with a rationale and learned shop preferences; the shop's own products; prepress profiles (preflight thresholds as a press profile, five built in); preflight: safe zone, total ink coverage read from the content stream, black build, trapping, spot colours, layers, image codecs; creep compensation from grammage and paper class.
1.5.9	The size dropdowns follow the template (the sheet you pick is the sheet you get); sheet-name matching with the orientation suffix; *Optimize N-up* can also shrink a layout, using the same arithmetic as validation.
1.5.10	Consecutive-page ganging (`GangingPageMode`); page order by dragging thumbnails, held as a permutation and re-applied after every document rebuild; `{loc:Loc}` fixed so language switching

	works across the whole interface; in-app help rewritten in both languages.
1.5.12	Native PDF/X-1a and PDF/X-3 packaging (`PackageAsPdfX`) — Ghostscript stops being a dependency; inert transparency constructs removed and files whose transparency paints refused (`NativeTransparencyFlattener`); the written file verified for transparency and, for X-1a, for DeviceRGB; ink-coverage scanning and Ghostscript discovery split into `NativeInkScanner` and `GhostscriptLocator` .
1.5.11	Fold geometry derived from the product (`FoldGeometry` , `FoldSizeMode`) instead of from dividing the sheet; one shared panel-fit check; fold marks for the folding bindings; live sheet preview and live validation in the template manager; item size from the standard-size list; presets renamed to starting points and grouped; bleed detection in file dimensions and format-scaled matching thresholds in suggestions; fit check for the pitch-spaced Step & Repeat grid.
1.5.3	Preview tab: uncapped thumbnails with a sliding memory window and page edits (rotation, grayscale natively on the engine); File → Save PDF; no auto-template at startup; tabbed file info; source-file pill (version/colours/min DPI); tolerant Flate decoder.

The complete, detailed changelog is in the `changelog.html` file bundled with the application.
