

IndexColorExtractor

Index color separation for screen printing

Dragonfly Colors

How this manual is arranged

I. Before you start

1. What index separation does and how it differs from halftone
2. Installation, license key, moving to another computer
3. Preparing the file: sRGB, resolution vs mesh, mode, transparency

II. One job, step by step

4. Extracting the colors — three modes and how to choose
5. Underbase and garment color
6. Reducing to the number of screens
7. **The reduced palette — what the swatches tell you**
8. The indexed document: Amount, dpi, two passes
9. Channels: channel solidity, registration marks, choke

III. Accuracy

10. The ink database and measuring opacity by eye
11. The error map and the texture loss map

IV. When something goes wrong — symptom → cause → what to do

Glossary at the end.

---

1. What index separation does

The artwork is reduced to a finite number of colors — exactly as many as you have screens — and from that moment every pixel is **one specific ink**. There are no pixels in an in-between color, because there is no in-between ink. Shades that are not in the palette are produced by **mixing dots**: dots of two inks sit side by side and the eye, from a distance, sees a single color.

How the dots are placed

This is done by **Floyd-Steinberg dithering**. For every pixel it picks the palette ink closest to the original, and whatever it failed to deliver — the difference between the original color and the ink color — it **pushes onto the neighboring pixels** it has not processed yet. The largest share, 7/16, goes to the right, the rest goes down.

This way the error is never lost, it is spread around the neighborhood. A single pixel may be too light or too dark, but **the average over an area matches the original** — and that is the whole halftone mechanism. The strength of that pushing is controlled by **Amount** (chapter 8).

Serpentine

Since almost half of the error goes to the right, reading every row from the left makes the error always travel the same way. In smooth areas it accumulates and produces **diagonal streaks**, all leaning the same way.

Serpentine reads every second row from the right, with a mirrored kernel. The drift then goes right, then left, and **cancels out between rows**. The grain becomes directionless. It costs nothing — neither memory nor time — and it is on by default.

How this differs from halftone separation

In halftone separation — simulated process — the halftone comes from **changing the size of the dot**: darker means a bigger dot, lighter means a smaller one. All the inks share **the same angle, the same line count and the same dot shape**.

In index separation the dot is always the same size — the size of a pixel — and lighter or darker comes from **how densely those dots lie**.

Where moiré comes from

From laying the **halftone over the threads of the mesh**.

In halftone separation the regular grid of dots interferes with the density and thickness of the mesh threads. In index separation exactly the same mechanism is at work: what is irregular is the distribution of **colors**, but the dots themselves lie on a **regular grid of pixels**. If its pitch gets close to the pitch of the mesh, the two patterns start to overlap.

There is no rule to it: sometimes moiré appears despite an indexed separation, because the pattern of the pixels has lined up with the pattern of the mesh. It is chance, not a particular pairing of resolution and mesh.

What follows from all this in practice:

- **Everything depends on the choice of palette.** In halftone separation an error in choosing a color is blurred into the mix of inks. Here a pixel is either one ink or the other — there is nothing in between. That is why most of this application and most of this manual is about one thing: **which inks to choose**.
- **The resolution of the file becomes a printing parameter.** An index dot is the size of a pixel, and the smaller the dot, the harder it is to wash out of the emulsion — chapter 3 covers this.
- **Registration has to be good.** Dots of neighboring inks lie next to each other, not on top of each other. A screen out of register shows immediately.

This application does three things: it extracts a palette from the artwork, it lets you bring that palette to a state that is workable with your inks and your press, and finally it splits the result into channels ready for film.

---

2. Installation and license

Requirements

Photoshop 24.0 or newer, Windows or macOS. The application runs in a panel — you open it from the **Plugins** menu.

First run

On the first start the application asks for a **license key**. You receive it by email after the purchase. Paste it, click activate and the panel opens.

How the key is checked

Over the internet — at activation, and then **once a week**. Between checks the application works without a connection and asks for nothing.

If there is no network at the moment of a check, **the application says so plainly** — it does not claim the key is bad — and **keeps working for another week**. Only if it fails to connect even once during that week does it lock until the key can be checked.

That distinction is deliberate: "no connection" and "subscription expired" are two different situations and the application names them separately. In the first you check the internet, in the second you pay.

The key is not deleted when a subscription expires. Once you settle the payment, the application unlocks itself, with no need to write to the seller.

Moving to another computer

The license is tied to a computer. To move it to another one, open **settings** (the gear) and click **Move to another computer**. That releases the seat — you then activate the same key on the new machine.

Do this **before** you remove Photoshop or reformat the disk. Recovering an occupied seat afterwards requires getting in touch.

What the application brings with it

Along with the application you get an **ink database** — measured opacity and behavior of several dozen inks on fabric. It is a starting point, not an oracle: your inks, your screens and your press will give different numbers. You can extend the database, correct it, or throw it away and start your own — chapter 10 describes how measuring works.

The database is loaded **once**, at the first start. If you delete it, it will not come back at the next start nor when the application is updated — your measurements will not be overwritten.

Where your data lives

- **settings and ink database** — in the application data folder, saved automatically
- **saved palettes** — `.dfpal` files, wherever you save them
- **database backup** — a `.dfink` file, via **Save database to file** in settings

Both are plain JSON inside. It is worth exporting the database every now and then — the application data folder is lost when Photoshop is reinstalled.

---

3. Preparing the file

Three things need checking before you click anything. Each of them can spoil the whole job in a way you will only see on the shirt.

Color space: sRGB

The application computes all color distances in sRGB. If the file is in Adobe RGB or ProPhoto, the same hex numbers mean something different from what the application assumes — and an ink number pasted from outside all the more so.

The application detects this and shows a yellow warning under section 1. If you see it, do **Edit → Convert to Profile → sRGB** before separating. The conversion recomputes the pixels so that the appearance stays — the artwork will not run away from you.

Resolution vs mesh

An index dot is the size of a pixel. How small it can be does not depend on whether it will hold on the screen — the emulsion will hold a dot smaller than a mesh opening too. **The limit is at exposure and washout.**

A very small dot on the film is too weak a stencil for the emulsion: light undercuts it, it hardens partly and does not come away during washout.

That is why 220 dpi is the ceiling for index separation — and already at 220 you need a CTS unit, printing wax straight onto the emulsion. A film held against a coated screen is too weak a medium for a pixel this small. Above 220 dpi you do not go at all: it will not wash out.

The mesh is matched to the resolution like this:

resolution	pixel	mesh for the colors	mesh for the underbase
160 dpi	0.159 mm	120/cm (305/inch) or 90/cm (230/inch)	77–90/cm (196–230/inch)
180 dpi	0.141 mm	120/cm (305/inch)	90/cm (230/inch) or 120/cm (305/inch)
220 dpi	0.115 mm	130–140/cm (330–355/inch)	130–140/cm (330–355/inch)

The underbase has its own column for a reason. It carries the same pixel as the colors, but where possible it goes on a mesh one step coarser: the white layer comes out thicker, whiter and more opaque. On the same mesh as the colors the underbase comes out greyer, and the colors sink into the fabric more and lose saturation.

The coarser mesh comes with one condition: **the shop has to be able to wash the pattern's pixel out of it**. If 180 dpi washes out on 90/cm (230/inch), the underbase goes on 90; if not, on 120/cm (305/inch), the same as the colors. At 160 dpi with the colors on 90/cm, the underbase steps down the same way, to about 77/cm (196/inch).

At 220 dpi there is nowhere to step down to. The pixel is so small that it is barely visible on the film, and even on 140/cm it is hard to wash out — the underbase goes on 130–140/cm (330–355/inch), the same as the colors.

There is no single floor — the mesh sets the resolution. A screen wants roughly 1.3 to 1.7 mesh openings across one pixel, and that one rule gives a different band for every mesh: about **134–176 dpi** on 90/cm (230/inch), about **179–234 dpi** on 120/cm (305/inch). 180 dpi is the bottom of the band for a 120/cm screen, the mesh most shops reach for.

160 dpi is the value that fits widest, and the one the application defaults to: it is the only resolution that sits comfortably on a 120/cm (305/inch) screen and on a 90/cm (230/inch) screen alike. If a shop prints at 55 lines per inch and goes no finer, 160 is right for it; for more margin on the coarser mesh, drop to 150 or 140. Work below the band for your own mesh and the ink still transfers perfectly well — what you give away is detail, because the pixel itself grows.

Yellow mesh or white. A white thread scatters light under the film and the edge of the dot always gets slightly exposed — the opening in the emulsion comes out smaller and the dot prints finer than it is on the film. A yellow thread does not scatter at all, so for raster work it is simply the safer one. The two also differ in exposure time: white mesh exposes faster, yellow takes longer.

Interference. Sometimes moiré appears on the print even though the separation is indexed: the pattern of the pixels has lined up with the pattern of the mesh. There is no rule to it — it is chance.

Mode and transparency

The document should be in **sRGB, 8 bit**, and the artwork **flattened** — text layers and effects turned into pixels.

Wherever there is no print, there should be transparency. The application skips transparent pixels at every stage: they do not enter the palette, they get no ink and they do not count in the diagnostic maps.

If instead of transparency the background has **any color** — white or anything else — that color becomes an ordinary palette color and will take a screen like any other ink.

To stop that from happening, you have to **set it as the garment color** (section 3) and mark in the box that this area is to be left **unprinted**. It then does not enter the palette, because the application treats it as the shirt rather than as an ink.

With a transparent background that question never comes up, because there is no background color in the palette to match against the shirt.

---

4. Extracting the colors

This is section 1 of the panel. What you will be working with later is decided here.

Analyze design flatness

The button computes how much of the surface is covered by the fifteen most frequent colors. You get the result as a percentage, together with the number of distinct colors in the file.

A high value means artwork with flat fields — a drawing, a logo, a vector illustration. **A low one** means painting and gradients — a photograph, a digital painting, soft transitions.

This number **does not say which mode will win**. It describes the artwork, it does not advise. The three modes are cheap — extract with each of them and compare, instead of guessing.

The three modes

Preserve accents — makes sure small, saturated patches do not disappear into the averaging. A small red light on a gray background then has a chance of reaching the palette. The default choice.

Dominant colors — picks the colors covering the most surface. Good when the artwork has large, calm fields and the detail matters less than the fidelity of the big areas.

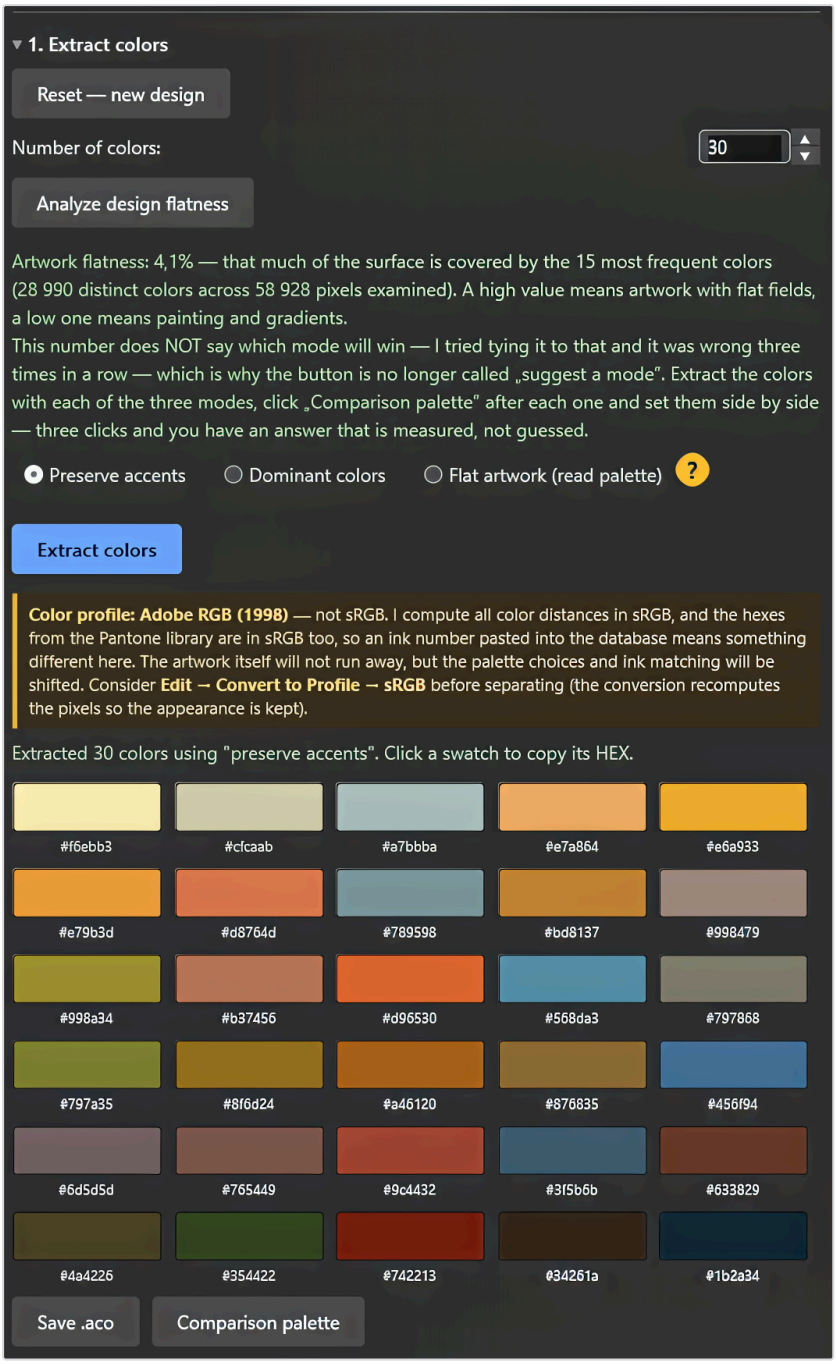
Flat artwork (read palette) — averages nothing, it simply **reads** the colors that are actually in the file. For artwork that already has a finite number of colors. On a photograph it will return random pixels.

Number of colors

Enter **more than you have screens** — thirty with ten screens is a sensible start. Section 4 will reduce it to the number of screens, and the surplus gives it something to choose from. Too small a number at this stage cuts off colors you will not get back later.

What you get

A grid of swatches with hex codes. Clicking a swatch copies its hex to the clipboard.



Section 1 — the flatness check, the three modes, the profile warning and the grid of extracted colors.

Save .aco exports the palette to a Photoshop swatch file. **Comparison palette** opens a window where you set the results of the different modes side by side — three clicks and you have an answer that is measured, not guessed.

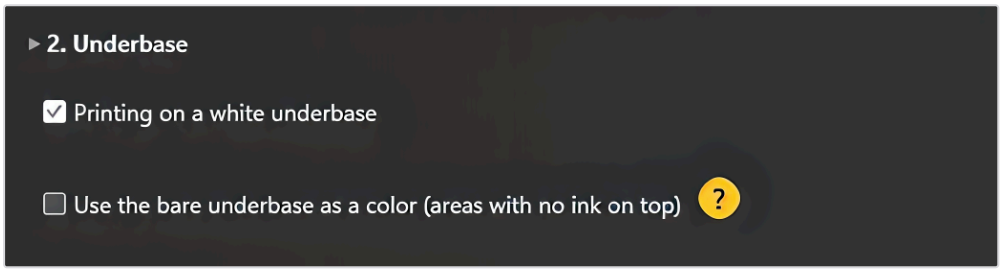
---

5. Underbase and garment color

Sections 2 and 3. Short, but they affect everything section 4 will compute.

Printing on a white underbase

Tick this if white goes under the colors. The application then stops assuming the ink lies straight on the fabric, and computes every color as it will come out on white.



Section 2 — both checkboxes.

Use the bare underbase as a color — appears once the box above is ticked. It lets you use the bare underbase as one of the colors of the artwork, in places with no ink on top. It does not take an extra screen, because the underbase is printed anyway.

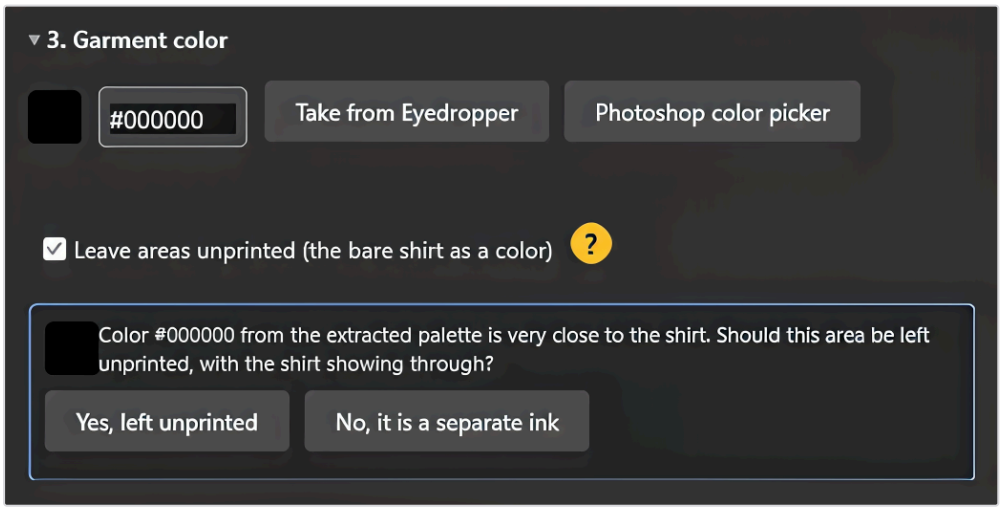
Garment color

Type a hex, take it with the Eyedropper, or pick it from the Photoshop color picker. The application uses this color to compute how the inks will come out on bare fabric.

Leave areas unprinted — treats the garment color as one of the palette colors. Wherever the artwork has the shade of the shirt, no ink will go.

If the extracted palette contains a color very close to the shirt, the application will ask about it itself:

Yes, left unprinted or **No, it is a separate ink**.



Section 3 with the question about a color close to the shirt.

When to use an underbase and when not to

On a **dark shirt** an underbase usually pays off: you get black for free, without spending a screen, and the shadows come out smooth.

On a **light or colored** one, be careful — dark shadows can fall out of the print and leave a hole in the artwork. Chapter 7.6 describes what the application does then.

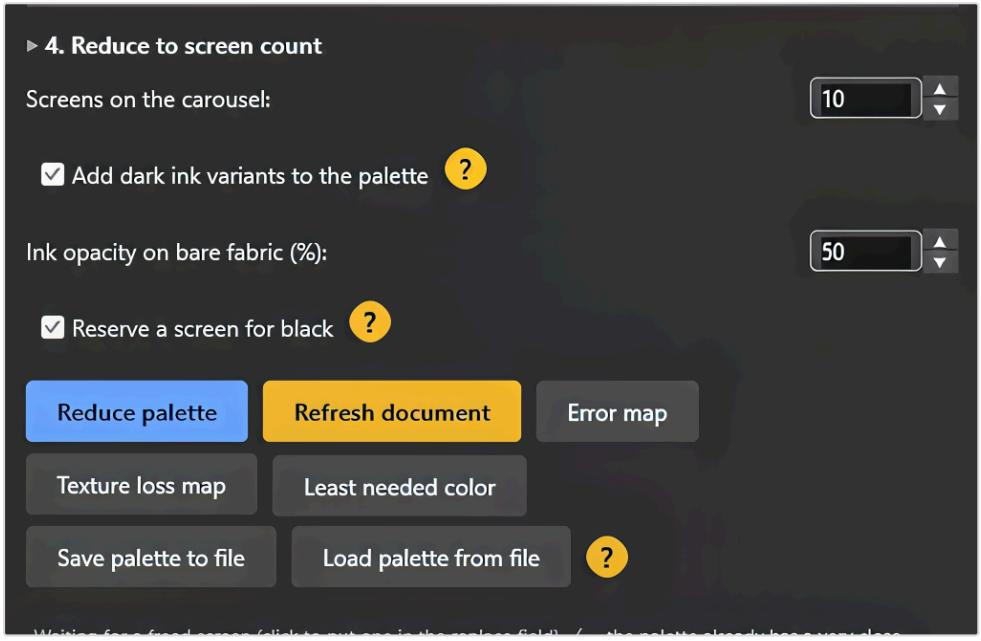
---

6. Reducing to the number of screens

Section 4, the first half. This is where thirty colors become ten.

Screens on the carousel

Enter as many as you physically have. The number covers **everything**: the underbase, the reserved black, every ink.



The top of section 4 — screen count, variants, opacity on bare fabric, black reservation.

The yellow line under the buttons sums the result up plainly — `Done: 9 ink colors + 1 underbase = 10 screens`. That is the number of screens you mount on the press.

Add dark ink variants to the palette

Ticked: every ink can produce **two entries in the table** — a light one on the underbase and a dark one straight on the fabric. Two entries, but **one screen**: what separates them is the underbase, which lies under the light variant and not under the dark one.

This lets ten screens produce a dozen-odd shades. Chapter 7.4 covers it in detail.

Ink opacity on bare fabric

The value the application assumes for inks that are not in the database. 50% by default.

This is a **starting assumption, not a measurement**. Inks from the database have their own opacity and do not use this value — you will recognize them by the green caption under the swatch.

Reserve a screen for black

Sets one screen aside for black and makes sure it stays black. Useful with outlines and text, where a black built from dots of several inks looks dirty.

Reserved black gets a yellow frame in the palette and the caption `(black from the reservation)`.

Reduce palette

The main button. After clicking it you get the reduced palette — and **from this moment you are working in chapter 7**, which describes every caption and every button next to the swatches.

The other buttons

Refresh document — recomputes the indexed document after a change to the palette. It lights up amber when the document is no longer consistent with the palette.

Error map and **Texture loss map** — diagnostics, chapter 11.

Least needed color — points at the ink whose removal will hurt the artwork least. For when you have to go down by one screen.

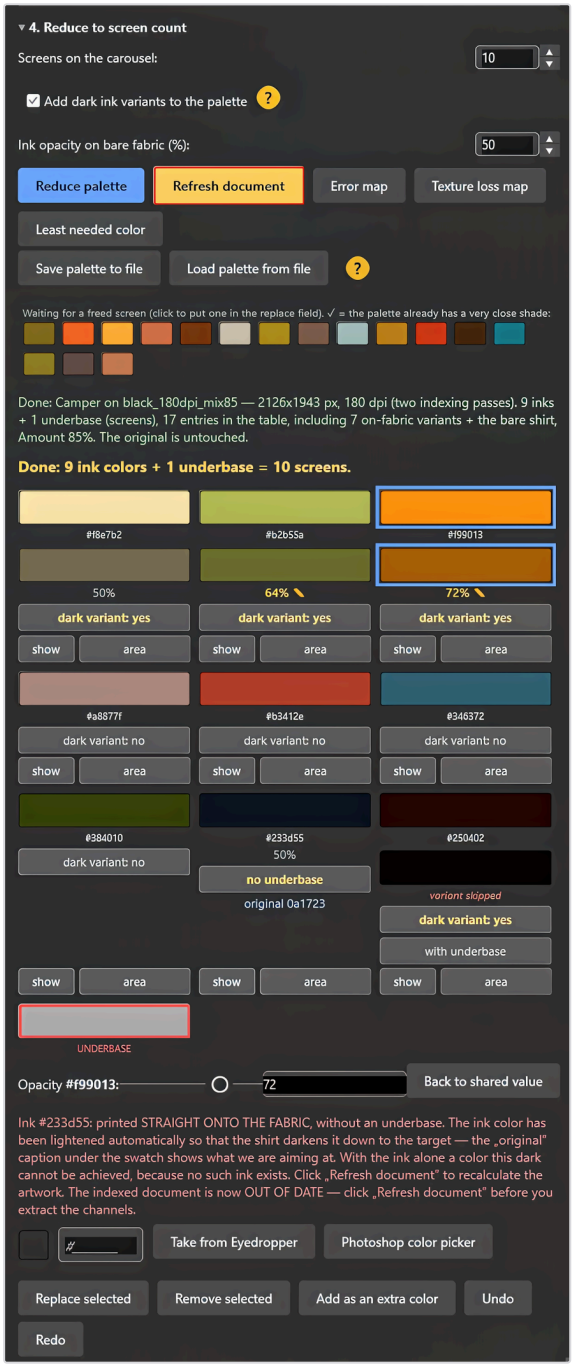
Save palette to file / **Load palette from file** — `.dfpal` files. Loading restores the separation down to the dot, together with the opacity of every ink and the area restrictions. Loading does not filter by extension, so it will also show files saved without one.

---

7. The reduced palette — what the swatches tell you

This is the densest place in the whole application. Under every color swatch there can be several captions, and next to it two or three buttons. None of it is decoration: each one answers a different question you ask yourself while choosing inks.

This chapter goes through them in order. It is worth reading once in full, and then coming back to the cheat sheet at the end.



The reduced palette in full — swatches, captions, buttons and the waiting queue at the top.

One thing to start with, because it helps everywhere: **almost every element of this section has a tooltip**. Hovering over a swatch, a caption or a button shows what a given number means and where it came from.

7.1. Where the inks in the palette come from

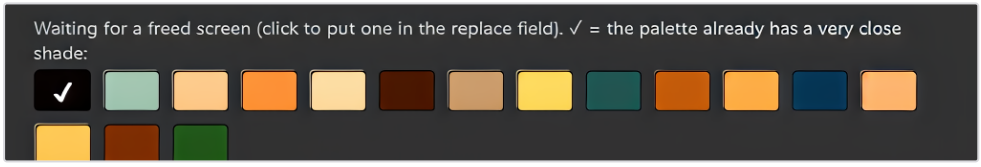
Step 1 extracts as many colors from the image as you ask for. Step 4 reduces them to the number of screens your carousel has. The ones that did not fit **do not disappear** — they land in the **waiting queue** above the palette.

The waiting queue has two sources and the application tells them apart.

"dropped during reduction" — the color was in the palette from step 1, but step 4 could not fit it into the number of screens. This is the most common case.

"did not fit during extraction" — this group appears **only in the "Flat artwork (read palette)" mode**. That mode averages nothing: it reads the colors that are actually in the file and picks as many of them as you asked for in "Number of colors". The losers do not vanish — the application remembers the twelve largest by area and keeps them here, so that you can pick them up by hand.

In "Preserve accents" and "Dominant colors" this group is always empty. There the palette colors come from averaging pixels rather than from choosing among ready-made ones — so there are no "losers" that could be pointed at.



The waiting queue. A checkmark on the first swatch means the palette already has a very close shade.

On screen this is one row of small swatches, not two. The split by source is internal — the swatches look the same and carry no visible captions. If the same color lands in both groups, it shows once.

Hover over such a swatch and you will see the lot: the hex, which group it comes from, how many pixels of the artwork it covered, which palette ink lies closest to it and by how much ΔE. When that distance is below the threshold, the tooltip adds that a separate screen will add almost nothing.

Clicking puts the color into the **replace field** under the palette. The status says what happened, and when the palette already has a very close shade it **warns plainly** and gives the ΔE, with a note that such a color only makes sense as a replacement for that one, not as an addition.

Movement both ways

This is the heart of working on a palette:

From the queue into the palette. Click a swatch in the queue — it goes into the replace field under the palette. Now press **"Replace selected"** and it takes the place of the selected ink.

From the palette into the queue. Remove an ink from the palette and it **returns to the waiting queue** — the status says so plainly: "— *it went back to waiting for a freed screen*". Nothing is lost as long as the color comes from the artwork.

> Exception: a color **picked by hand from the color picker** has nowhere to return to — it was never in the artwork. The application then says *"(color from outside the artwork — gone)"*.

This makes the palette **exchangeable**, not final. You can throw an ink out, see what happened to the artwork, and put it back, because it has been waiting above the palette the whole time.

Outlines in the queue

When you take a color with the eyedropper, the queue shows whether one like it is already waiting for a screen:

- **blue outline** — this is practically the same color
- **yellow outline** — close, but recognizably different

A ✓ on a swatch means the palette already has a very close shade — a separate screen will add almost nothing.

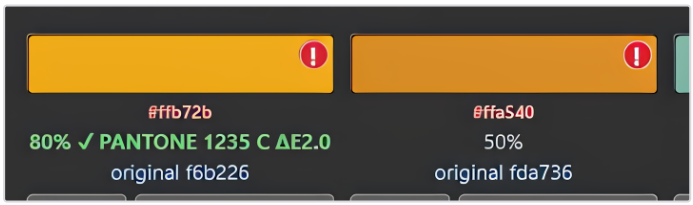


A yellow outline: a color close to the one taken with the eyedropper, but recognizably different.

7.2. The caption under the swatch — where the opacity came from

Under every ink stands its opacity. **The color of that caption tells you what the number is worth**, and that is the most important piece of information in the whole palette.

Green: `70% ✓ PANTONE 186 C ΔE2.3` — from the ink database



The green caption: opacity taken from the ink database, with the name of the ink found and the ΔE distance.

The application found an ink in your database lying close to this color and took **its** opacity. Behind the number stands the name of the ink it found and **ΔE — the distance from it**.

ΔE tells you how far you can trust that number:

ΔE	what it means
below 1	practically the same color
1–2	you only see the difference with the two side by side

ΔE	what it means
above 2	**you will not see this caption at all** — that is the matching threshold

In other words: the green caption appears only when ΔE is below 2. Above that the application decides it is a different ink and does not take its opacity.

Why you need that number. "70% ✓ PANTONE 186 C" without the ΔE would read like an identification of the ink — and it is only **the closest entry in your database**. With a ΔE near the threshold it is worth checking whether this really is the ink you are going to mix, or only something similar.

Yellow: `70% ✎` — set by hand



Gray 50% is an assumed value, yellow 72% ✎ one set by hand.

You moved the slider on this ink yourself. **The shared "Ink opacity on bare fabric" field will not overwrite it** — your decision outranks the collective setting.

Gray: `50%` — assumed

This ink is not in the database and there was nothing to take. This is a **starting value, not a measurement**. Measure this ink in the settings, or correct it with the slider, before you trust the result.

Red italic: `variant skipped`

The dark variant of this ink did not make it into the table, because **at this opacity** it is indistinguishable from another entry — and since both would give the same color, the second would add nothing. The tooltip gives the hex of that entry.

It is usually **the same ink on the underbase**: the ink covers strongly enough that straight on the fabric it looks the same. But the application compares against **the whole table**, so it happens that the variant falls close to the appearance of **another ink in the palette**. That second case matters more than it looks: if such a variant stayed, dithering would start mixing the dots of two different screens and the shirt would come out dusty.

Lower the opacity and the variant comes back. A lower opacity lets the color of the shirt through, which darkens the ink on bare fabric and moves it away from the version on the underbase. How far you have to go down depends on the ink and on the color of the shirt — sometimes a dozen-odd points are enough, sometimes you have to go much lower.

> **Careful, this is the opposite of the situation with the red `!`.** There the ink does not reach its target and the opacity has to be **raised**. Here the ink covers too strongly and the opacity has to be **lowered**. Both captions concern the same slider, but they pull it in opposite directions.

The slider describes the ink, it does not drive the press

By lowering the slider you are telling the application that this ink covers less — and the artwork is recomputed as if that were so. The distribution of dots on the films changes. The ink in the bucket stays the same.

For the print to agree with what you see on screen, the opacity has to be lowered **on the press too**. That is done with **a finer mesh for that one ink**: the same ink through a 120 mesh will leave a thinner layer than through a 90 and will cover less. That ink's channel then goes onto a different screen from the rest.

There is also a second route, which removes the problem altogether: **enter the ink into the database** (chapter 10). A measured ink has its own opacity and the application takes it by itself — you then do not have to touch the slider at all, because the default value is only an assumption for inks the application does not know. The caption under the swatch says plainly where the number comes from: green means a measurement from the database, yellow your own manual setting, gray an assumption.

Green: `underbase 80% → screen freed`

This is not a warning, it is an **opportunity**. At this underbase opacity the bare underbase will start to impersonate this ink and it will no longer be needed — **you free one screen**.

What matters most is that a more opaque underbase **costs no extra arm**. It is the same stencil and the same pass; the thicker layer is added on the press:

- with **a second stroke of the squeegee**,
- with **a flatter squeegee angle** — a shallower angle lays a thicker layer,
- with **less pressure** on the squeegee.

So the screen you free is bought with a press setting, not with another color.

7.3. The caption next to the hex — where the ink came from

In the caption itself, right after the color code, there may be a note:

- `(manual)` — a color chosen by you, not computed from the image
- `(black from the reservation)` — a black inserted by the "Reserve a screen for black" option

Both are **locked**: recomputing the palette will not touch them.

7.4. The ink on the underbase and the ink on the fabric

The same ink looks different printed on a white underbase (light) and straight on the material (dark). When you turn on **dark variants**, a **second swatch** appears under every color — the lower one shows how the ink will come out on bare fabric.

This does not cost another screen. One ink gives two entries in the index table; what separates them is the underbase, which lies under the light variant and not under the dark one.

Under the lower swatch stands **the opacity figure** — that is, how much of the shirt shows through this ink. Here we are back to chapter 7.2: the color of that figure says whether it is measured or assumed.

The `dark variant: yes / no` switch turns this on and off for a single ink.

7.5. The `with underbase / no underbase` switch

This is a decision taken **separately for every ink**: whether this particular ink is to land on the underbase or straight on the material.

Why this button exists at all

It came out of a specific problem: **a very dark color that has no equivalent among inks**. The darkest thing you can mix is sometimes clearly lighter than what stands in the artwork — and an ink laid on a white underbase shows exactly itself, so such a color would come out falsified. The dark parts would turn grayish and the whole artwork would lose its depth.

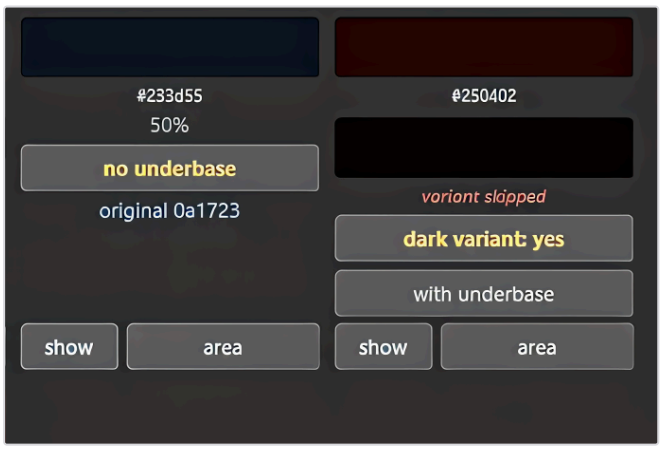
Straight on the fabric the same ink **is not lightened by the underbase** — the dark material darkens it and pulls it to where it was supposed to be. That is why with dark colors this switch is worth considering, instead of looking for an ink that does not exist.

What happens after switching to "no underbase"

The ink no longer lies on a white underbase. What you see is **a mixture of the ink with the material**, so the color would come out darker and less saturated than it was in the artwork.

The application prevents this: **it picks an ink lighter by exactly as much as the fabric takes away**. After printing, the mixture is meant to give the color from the original.

You will recognize it by the **blue caption** under the swatch:



An ink switched to “no underbase”: the blue original shows the target, the swatch shows the result on the shirt. Next to it, a swatch with a skipped dark variant.

`original 1a2b3c`

This is the **target** — the color that stood in the artwork and that we are aiming at. The swatch itself shows how it will come out on the shirt, and **the hex caption above it is the new ink color** — the one you have

to mix.

Three different things in one place, so it is worth remembering:

what you see	what it is
the swatch	how it will come out on the shirt
the hex under it	the ink color you will mix
`original ...` in blue	the color we are aiming at

Careful: the red `!` can appear here too

Switching even one ink to "no underbase" triggers the same lightening as the **"Lighten for fabric"** button. If such an ink would have to be lighter than anything you can mix, it gets a **red exclamation mark**. Chapter 7.6 describes it — that is where this case happens most often, because it concerns the whole palette at once.

7.6. Printing on a light shirt without an underbase

On light material an underbase is often unnecessary — and then **the whole palette** goes straight onto the fabric. The button for this is **"Lighten for fabric"**, above the palette.

It does the same thing as the switch from 7.5, only **for all the inks at once**: it picks them lighter by as much as the material takes away.

After clicking:

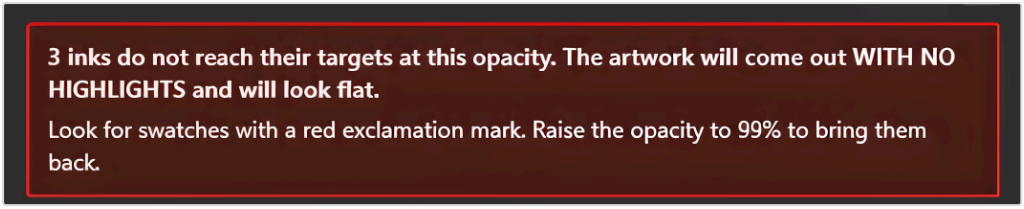
- a blue `original ...` appears under every swatch
- the status says how many inks ran up against white (if any)
- **click "Refresh document"** to recompute the artwork

The lightening does not accumulate. Every color remembers its shade from before the compensation and is always computed from that. Changing the shirt color or the opacity recomputes them from scratch, and the **"Undo lightening"** button restores the colors read from the image.

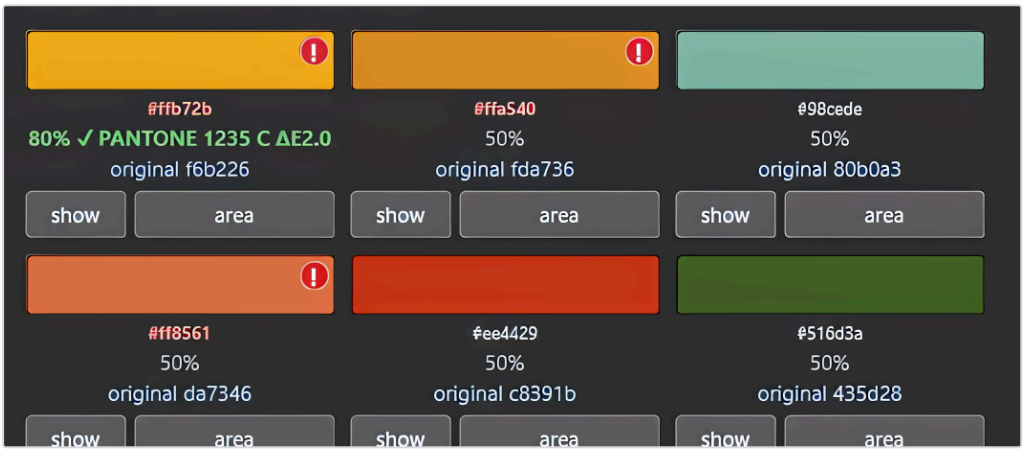
When an ink does not reach its target — the red `!`

Not every color can be reproduced this way. Colors close to white would need an ink **lighter than white**, and there is no such thing.

Such a swatch gets a **red exclamation mark in the corner**, and its caption turns salmon.



The warning above the palette.



Three inks with a red exclamation mark and a salmon caption. The first has its opacity from the database, the others assumed.

A line appears under the palette: "*2 inks do not reach their targets at this opacity. The artwork will come out WITH NO HIGHLIGHTS and will look flat.*"

What to do about it, in order:

1. **Raise the opacity of that ink with the slider.** The warning says how much is needed: "*At least 68% is needed*". The more densely the ink covers, the less the fabric darkens it and the higher it reaches.
2. If the application says "*This target cannot be reached at any opacity*" — that color needs **an underbase or white**. Switch that ink back to "with underbase" or add white.



The same artwork after raising the opacity — the exclamation marks are gone.

What the application will NOT do: it will not shift the hue. To an ink that runs up against white it gives back **lightness only, never hue** — it goes down all three channels at once, so the print comes out darker than the original, but in the same color and saturation. That is how it has to be, because dithering can dim a color, never saturate it: an ink that is too dark

> you will lighten with dots of a lighter neighbor, but an ink washed out of color will not be saved by any arrangement of dots.

7.7. The `area` button — restricting an ink to a part of the artwork

This is the most powerful tool in this section and at the same time the one needed least often. It is worth knowing it is there before you meet the artwork that needs it.

What it is for

Dithering distributes the inks over the whole artwork according to where they fit by color. Sometimes that is not enough:

- **a piece of text or a logo is meant to be solid**, not built from dots of several inks
- **one ink gets in where it should not** — a skin tone appearing in the sky, for instance
- you want to **lighten a fragment with white alone**, with no foreign colors falling in

An area is a mask: you tell the application where that ink **may** and where it **may not** lie.

Where the shape comes from

From an **alpha channel in the original**. You make a selection in Photoshop and save it with **Select → Save Selection**, then pick it from the list in the panel.

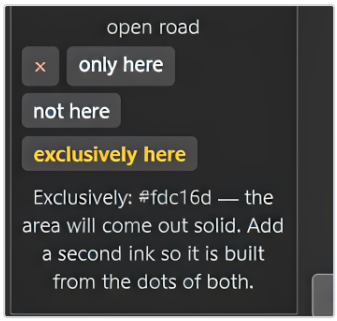
Why a channel and not the selection itself: a channel has a name, so the application reads it **again at every refresh of the document**. You correct the channel with a brush in Photoshop, click "Refresh" and the separation takes it into account — without clicking anything in the panel.

The three modes

mode	what it does
only here	This ink will be laid **only in this area**. Outside it, it will not appear at all.
not here	This ink **will not be laid in this area**. Nowhere else does anything change.
exclusively here	In this area **only the inks that marked it this way** are allowed.

The difference between **"only here"** and **"exclusively here"** is crucial and easy to miss:

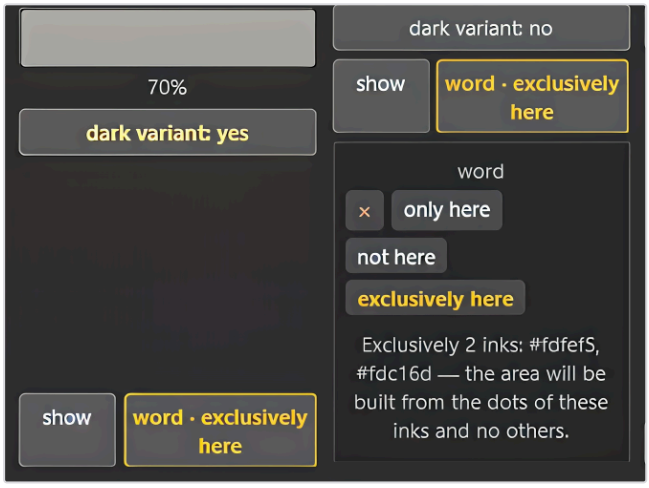
- **"only here"** restricts the **ink** — it says where the ink may lie
- **"exclusively here"** restricts the **area** — it says which inks may be used in it



The area picker opened: the three modes and a description of the effect.

How to make solid text: mark the area of the text on one ink in **"exclusively here"** mode. The area will be built from one ink, so it comes out solid.

How to make a highlight with white alone: mark **the same area** on a second ink, also in "exclusively here" mode. The area will be built from the dots of those two inks and no others.



The same area assigned to two inks — it will be built from the dots of those two and no others.

The rules work together

One ink can have **several area rules**. Then:

- **the prohibitions add up** — the ink will not enter any of the forbidden areas
- **"exclusively here" outranks the prohibitions** and wins in its own area

The button under the swatch shows the state: `area` (no rules), `channel-name · only here` (one rule) or `areas: 3` (several — details in the box that opens on click).

7.8. The other buttons

`show` / `hide` — blacks out the whole artwork except this one ink. With several similar shades this is the only way to establish **which ink serves which area**. With dark inks the cover is light, so that black does not disappear against black.

The status gives the **size of the area** along the way: *"Ink preview #b3412e — 47 200 px, that is 8.1% of the print*". The percentage is computed from the pixels actually printed, leaving out the transparent margin —

and that is the number that tells you whether a given ink earns its screen.

The underbase swatch stands apart and is labeled `UNDERBASE` or `UNDERBASE + COLOR` — the second means that the underbase itself was used as an independent color in the table, so it competes with the inks for pixels.

`black — no dark variant` on a black is not a fault. Black covers fully: on the underbase and straight on the fabric it looks the same, so a second entry in the table would add nothing.

7.9. Cheat sheet

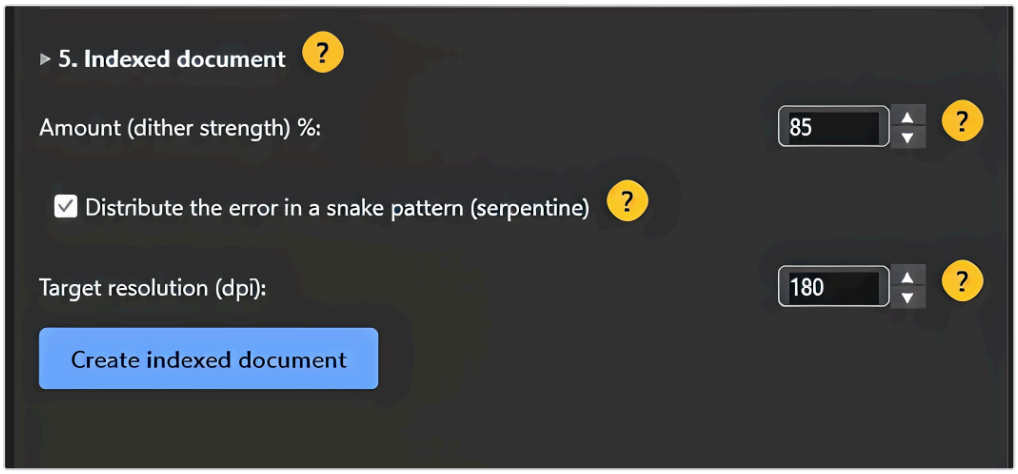
what you see	meaning
`70% ✓ PANTONE 186 C ΔE2.3` *(green)*	opacity from the ink database, ΔE = distance from the ink found
`70% ✎` *(yellow)*	opacity set by hand, the shared field will not overwrite it
`50%` *(gray)*	assumed opacity — this ink is not in the database
`variant skipped` *(red)*	dark variant indistinguishable from another entry — **lower** the opacity to bring it back
`underbase 80% → screen freed` *(green)*	at this underbase opacity you free a screen
`original 1a2b3c` *(blue)*	the target before lightening — the color from the artwork
`!` in the corner, salmon caption	the ink does not reach its target — raise the opacity
`(manual)`	a color chosen by you
`(black from the reservation)`	black from the "Reserve a screen for black" option
`dark variant: yes/no`	the second appearance of this ink — on bare fabric
`with underbase / no underbase`	whether this ink lies on the underbase
`area`	restriction of the ink to a part of the artwork
`show`	preview: what this one ink prints, plus the size of its area

---

8. The indexed document

Section 5. Creates a **new** document in which every pixel is exactly one of the appearances of the inks from your palette. The original is left untouched.

One ink is one screen, but it can give two entries in the table — a light one on the underbase and a dark one on the fabric (the `dark variant` switch under the swatch in section 4).



Section 5 — Amount, serpentine, target resolution.

Amount — the strength of dot mixing

85% by default. The recommended range is 75–90%.

Dithering picks the nearest table entry for every pixel and spreads **the difference** onto the neighboring pixels. Amount decides how much of that difference travels on: at 85% a fifth of it is thrown away.

In a large patch this loss accumulates and the area is pulled toward a single ink instead of being built from a mixture of two. **The symptom: a color lying between two inks comes out as the nearer one, with sparse dots of the further one** — instead of an even mixture of both.

When some area has the wrong shade, try **100%**. The mixing will be faithful, at the price of a more visible grain in flat places.

Changing Amount after the document has been created lights the **Create** button again and makes a **separate** document with that number in its name, next to the previous one. Two files with different mixing strength can therefore be compared side by side, not from memory.

Distribute the error in a snake pattern (serpentine)

Leave it ticked.

Dithering pushes **7/16 of the error — almost half — to the right**. When every row is read from the left, the error always travels the same way and in smooth areas it accumulates into **diagonal streaks** leaning the same way. Serpentine reads every second row from the right, so the drift goes right, then left, and cancels

out between rows. The grain becomes directionless, and the regular diagonal structure — which can interfere with the mesh of the screen — disappears.

It costs nothing: neither memory nor time. Turn it off only to see the difference.

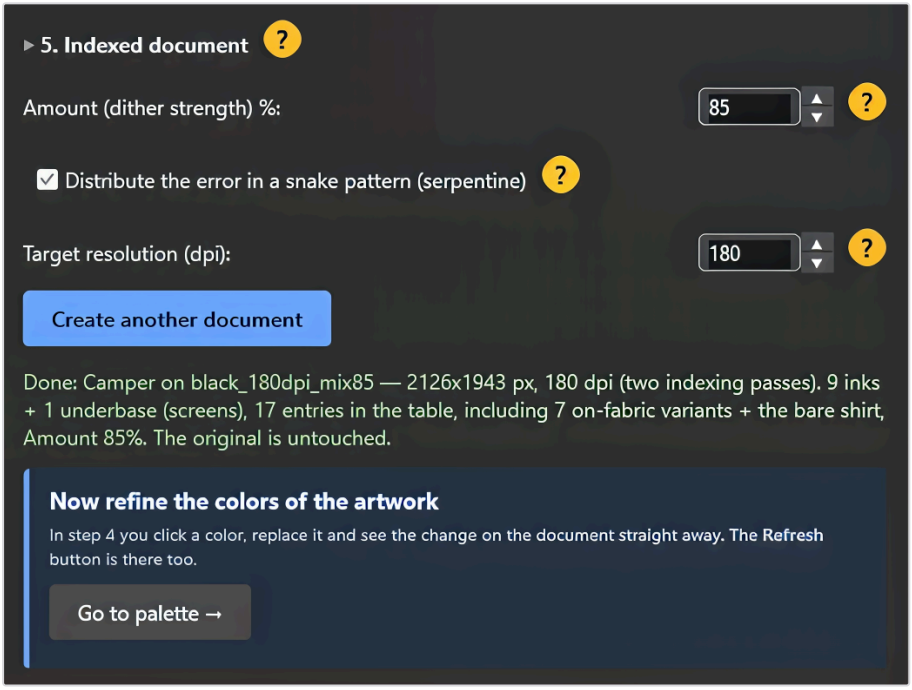
Target resolution and the two passes

The application indexes **first at the full resolution of the file, then downscales and indexes again**. The downscaling on its own squeezes blocks of color together and blurs hard edges, so the result is smoother than indexing straight at a low resolution.

Entering the same value as the file resolution skips the second pass.

For matching the value to the mesh — the table in chapter 3.

After it is created



After the document is created the button turns into “Create another document”, and the application offers to move to the palette.

The indexed document is **alive**: you go back to section 4, replace colors and click **Refresh document**. There is no need to create it again.

When the palette suits you, you move on to the channels.

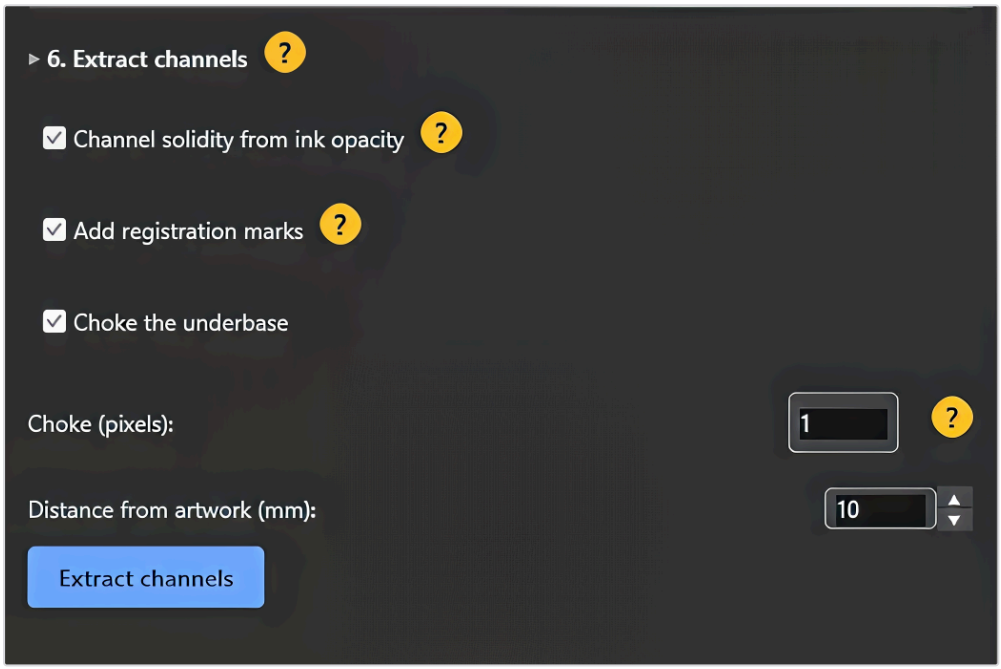
---

9. Extracting the channels

Section 6. Splits the indexed document into spot color channels — one per ink, plus the underbase channel.

One ink = one channel, even if it had two entries in the table: what separates them is the underbase, which lies under the light variant and not under the dark one.

The channels are numbered in the **suggested** printing order — the underbase, then from the darkest to the lightest. At the press the order usually needs adjusting.



Section 6 — channel solidity, registration marks, choke and distance from the artwork.

Channel solidity from ink opacity

This setting concerns the preview only. It has no effect whatsoever on the films. The separations are identical regardless of this value.

The solidity of a spot channel controls how Photoshop composites the channels on screen and in a composite print. **Do not confuse it with the ink opacity slider in section 4** — that one is a separation parameter and decides which pixels go onto which screen.

Ticked: every ink gets its own opacity, the same one the separation was computed with. The underbase and the shirt go to **100%**. The underbase not because white covers that well, but because its opacity has already been taken into account once — when deciding which inks need it. Entering it a second time here would count the same thing twice and the whole artwork would come out too dark.

The number in the channel will be different from the one on the slider, and that is how it should be. The slider gives *perceived* opacity, and Photoshop wants a physical value:

slider	channel
50	18
60	28
70	**41**
80	57
100	100

Without that conversion the ink in the preview would cover far more strongly than the separation assumes.

Unticked: all the channels get one value from the field below.

Registration marks

Six marks — a circle with a cross — three at the top and three at the bottom, in the middle and at both edges. The canvas will be enlarged, because the marks have to lie outside the artwork.

They go into **every** ink channel and into the underbase.

Distance from artwork sets how far from the graphic they lie. 10 mm by default.

Choke (narrowing the underbase)

The underbase will be printed smaller than the top inks by the given number of pixels, so that when the press is out of register it does not show as a white outline from under the color.

At 160 dpi one pixel is about **0.16 mm**; at 180 dpi about **0.14 mm**.

For flat artwork — a drawing with an outline and uniform fields. With artwork that has gradients and soft transitions, a choke needlessly thins the underbase in the midtones.

The registration marks are not choked — they have to be identical in every channel.

When the detail is too thin to choke. Taking one pixel off can eat a thin line down to nothing — there is nothing left in the file to shrink. Instead of choking it, take the same film you use for the color, expose it on a **white** mesh and **give it a longer exposure**: the scattered light undercuts the edge and the opening narrows by itself, smoothly, below one pixel. Here the white thread is the tool, not the flaw.

---

10. The ink database

Everything the application computes about the appearance of an ink rests on one number: **how much the ink covers on fabric**. An ink not held in the database gets the assumed value from section 4 — and that is guesswork. An ink from the database gets your measurement.

You open the database with the gear, section **Ink database**.

The database that ships with the plugin

The database arrives holding the author's own measurements. You can keep them, add to them, or wipe them and start your own — a newer version of the plugin will not overwrite your database.

Before you lean on them, you need to know how they were made:

- **120/cm (305/inch) mesh**,
- **one squeegee stroke**,
- **Wilflex Rio** inks,
- opacity read by eye, with the print held up against the screen.

If you print with different inks, this is not your data. You can wipe the database and build your own from scratch, but it is usually faster to go color by color, check whether your ink behaves the same, and correct only what differs.

This is not pharmacy work. So many things affect opacity at once that it cannot be set to the percent — the realistic accuracy is tens of percent. It shifts with pallet temperature, squeegee angle (more raked or more upright), squeegee durometer (a soft 60 against a hard 90), more or less off-contact, mesh tension, and whether the ink has been worked and warmed up or comes straight from the tub. Ten points either way is not an error — and nothing to worry about.

What the application shows

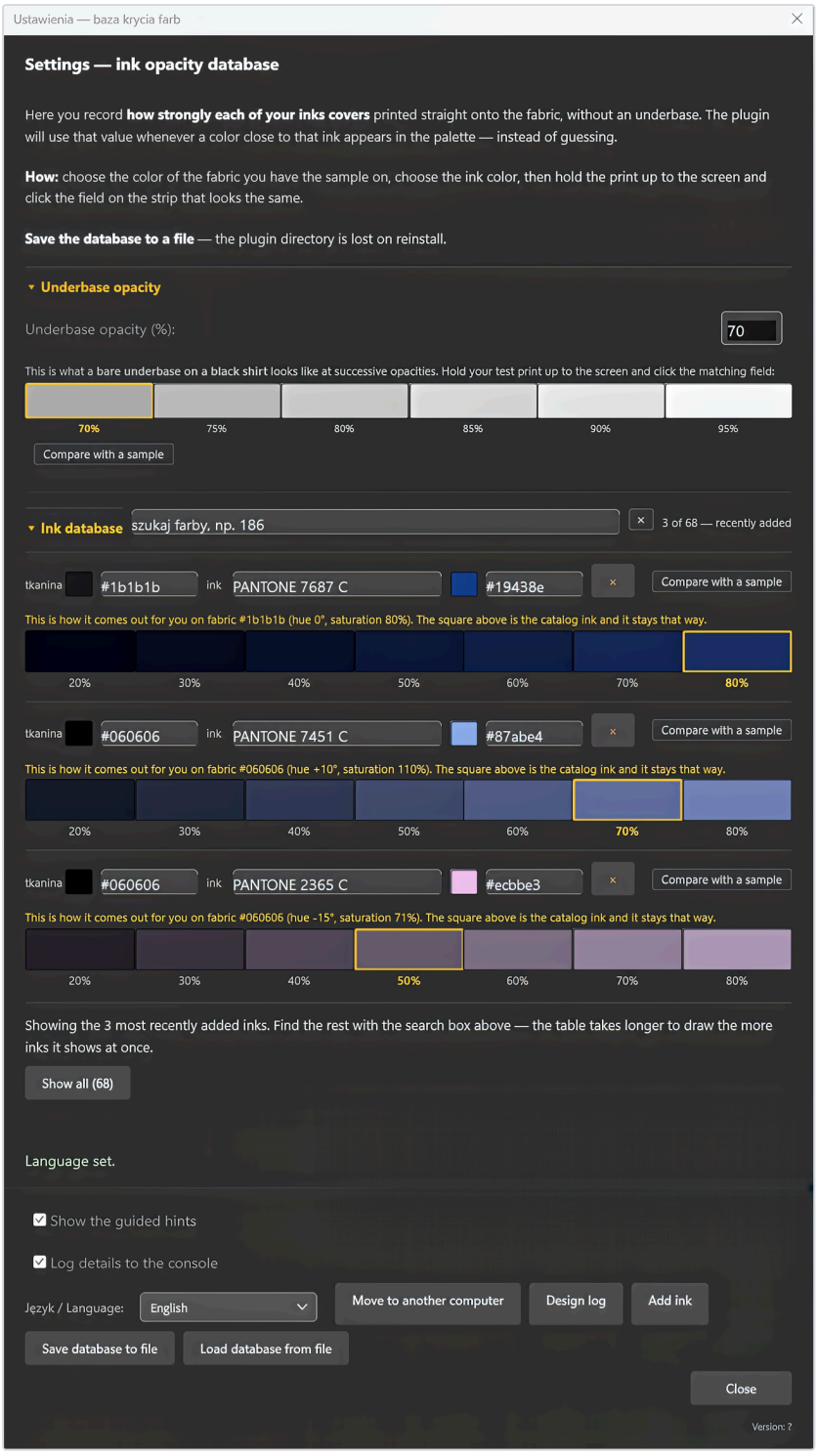
The table draws **the three most recently added inks** by default. The rest is reachable with the search box at the top, and the **Show all** button draws the lot.

This is not a limitation out of laziness: every row costs about 34 elements on screen, and UXP recomputes the layout of the whole window at every change of its size. With a few dozen inks the window starts to stall for several seconds; with two thousand it would not open at all. Shortening the list makes that cost constant.

How opacity is measured

You need **a print of that ink on the fabric** you are going to print on. The rest happens in the settings window, in the **Ink database** section.

1. First the fabric, then the ink



The settings window: the underbase opacity strip at the top, below it the ink database with the three most recently added entries.

The order matters, because everything the application draws from here on is computed relative to the material.

1. Click the **`fabric`** square in the ink row and pick the color of your shirt from the color picker. Not "black" out of habit — **tell a deep black from a broken one**, because it shows in the result.
2. Click the **`ink`** square and point at the color you are entering into the database.

3. Only once both are set, click `Compare with a sample`.

2. Conditions — with the sample already at the screen



The compare-with-a-sample window: the strip of textured fields, the hue and saturation tuning, and the light and screen settings.

A window opens with a strip of fields. **Take the print in your hand and hold it against the screen before you set anything.**

1. Set the **room light** — which of the four options you have over the table.
2. Set the **screen brightness**, looking at the sample. This is not a "once and for all" setting: inks differ in luminosity. A bright color needs the brightness raised for the screen to reach the sample at all; a dull one needs it lowered, so that the screen does not over-color it.

The background of the window is wide and in the color of the fabric for a reason: it is the **reference** against which you hold an unprinted piece of the shirt. If it were light, the eye would adjust to the panel and would read the same lightnesses differently.

3. Hue and saturation — on the 80% field

With the sample in front of the monitor, tune the **hue** and **saturation** to the **80%** field. Those two rows move the color sideways; the lightness stays the one from the strip above.

An ink on fabric rarely lands exactly where the catalog color is — it can be warmer, cooler or washed out. This is where you even that out.

4. Only now read off the opacity

When the color agrees, run your eye along the strip and look for the field at which **the border between the print and the screen disappears**. **Squint slightly** — that cancels the weave of the fabric and leaves the lightness alone, and the eye catches a vanishing border far more reliably than the similarity of two patches side by side.

The field you click writes the opacity into the database.

> **The textured fields are darker** than the smooth swatches in the table, and that is how it should be — a speckled surface always looks darker than a smooth one of the same lightness. Your print is the reference, not that swatch.

When no field fits

The opacity slider moves the color along a straight line from the fabric to the ink. If you run through the whole strip and **nothing** fits, it means the sample lies **off** that line — it is dirtier or of a different hue. This is not a matter of hitting the right value: go back to **hue and saturation** and nudge the color across to your print with them.

Texture

The fourth setting in this window, next to the light and the brightness: **how much the fabric eats the color of the ink**. `stronger` by default. It concerns how the application draws the fields — not the separation.

You set the fabric once

An ink is measured **on a specific fabric**, not in a vacuum — and you enter that fabric on the ink once, at the moment of adding it to the database. After that you do not touch it.

It does not have to be the same fabric you will be printing on. You can make the sample on dark navy for all it matters: if you mark it as navy, the application computes the rest and extracts a **percentage of opacity**, which is already a value independent of the material. That is what goes into the database and that is what works later in every separation.

> If you ever did change the fabric color on an existing entry: **the opacity stays**, only the **hue correction** is discarded — because that described the color of the ink on that material and would mean something else on another. It is a safeguard, not a step in the work.

Backing up the database

Save database to file creates a `.dfink` file with the date in its name. Do it every now and then: the application data folder is lost when Photoshop is reinstalled, when you change machine and when something breaks.

Load database from file replaces the current database entirely. The application says so plainly in the status.

---

11. The error map and the texture loss map

Two buttons in section 4. Each creates a separate diagnostic document — they touch neither the original nor the indexed document.

The error map

Shows **where the result departs from the original**. The stronger the red, the larger the color difference in that place.

What to look for:

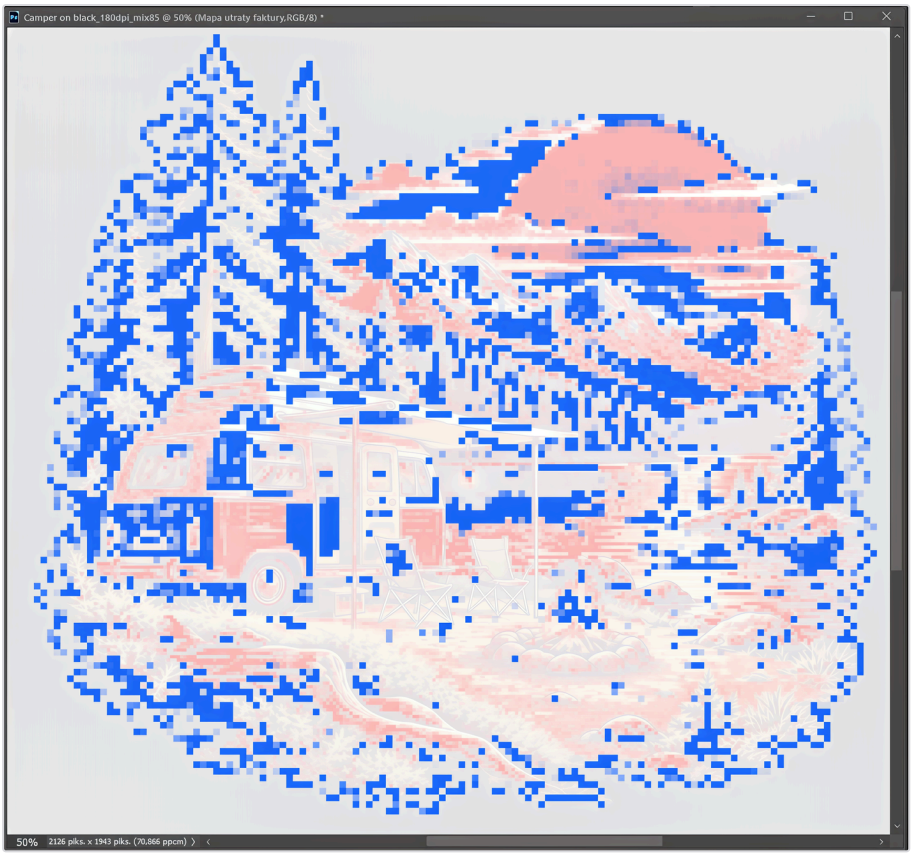
- **A large, evenly red patch** — the palette lacks an ink for that area. Consider adding a color or replacing the least needed one.
- **Red only on the edges** — normal. Dithering always has the most work where the color changes abruptly.
- **Red scattered evenly over everything** — the palette is too small for this artwork, or the Amount is too low.



The error map: the stronger the red, the larger the difference from the original.

The texture loss map

Shows **where tonal differentiation has disappeared** — places where several different shades of the original came out as a single ink. The blue fields are the areas where the artwork flattened out.



The texture loss map: the blue fields are the places where the artwork flattened out.

A large blue field means that dithering **had nothing to mix with** — one ink served that whole range on its own. There are two causes and they have different remedies.

A missing rung in the ramp. Between two neighboring inks there is a gap in lightness with nothing to fill it. The remedy: add an ink from **the middle of the gap**. It costs a screen.

An ink sitting in the middle of a wide range. An entry lying exactly in the middle captures every pixel of that range and has nothing to share the pixels with. The remedy: **move that ink** — lighten it or change its hue. Part of the range will then fall closer to a neighbor, dithering returns and the texture appears by itself. **This costs no screen.**

The second case is more common than it seems, and it is worth checking first — lightening one ink can unblock a whole area into which its neighbors immediately flow.

Read both maps **with the artwork open next to them**. On their own they are pretty, but without the comparison they do not say whether what lights up actually matters on the shirt.

When something goes wrong

symptom	cause	what to do
The hexes in the palette do not match what you see in Photoshop	the file is not in sRGB	Edit → Convert to Profile → sRGB, then extract the colors again
A large area has the wrong shade, with sparse dots of another ink	Amount too low — the difference accumulates in a large patch	raise Amount to 100% (ch. 8)
Smooth areas have diagonal streaks	serpentine turned off	tick **Distribute the error in a snake pattern**
The artwork comes out flat, with no highlights	the inks do not reach their targets at this opacity	look for swatches with a red `!` and raise their opacity (ch. 7.6)
A white outline of the underbase shows from under the color on the shirt	the press is out of register	turn on the **choke**, start with 1 pixel (ch. 9)
The finest dots do not come away when washing out the screen	the dot is too small for your film + emulsion + lamp	go down in resolution; 220 dpi is the ceiling, and that already with a CTS unit (ch. 3)
Moiré shows on the print despite an indexed separation	the pattern of the pixels has lined up with the pattern of the mesh — chance, not a rule	go down in resolution or raise the mesh (ch. 1 and 3)
Dark shadows fell out of the print and left a hole	printing on a light shirt without an underbase	ch. 7.6
A large patch lights up on the error map	there is no ink for that area	add a color or replace the least needed one (ch. 11)
A large blue field on the texture loss map	one ink serves the whole range on its own	first **move that ink** (lighten it or change its hue) — the neighbors will start flowing in; only if that does not help, add an ink from the middle of the gap
The channel solidity in Photoshop differs from the slider (e.g. 41 instead of 70)	that is how it should be: the slider gives **perceived** opacity ("looks 70% of the way to white"), and Photoshop wants a **physical** value	do nothing — the application converts it itself, conversion table in ch. 9

symptom	cause	what to do
The indexed document does not keep up with the palette	the changes in section 4 have not been recomputed yet	click **Refresh document** (it lights up amber)
A saved palette does not show up in the load window	saved without an extension	the window does not filter by extension — the file is there, look for it by name
The settings window stalls when resized	the whole ink database is being drawn	close it and open it again — it goes back to three rows (ch. 10)

Glossary

Amount — the strength of dot mixing in dithering. How much of the difference between a pixel and the chosen ink travels on to the neighbors. 100% = faithful, but a coarser grain.

Choke — narrowing the underbase relative to the top inks, so that it does not show as an outline when the press is out of register.

Dithering — spreading the error onto the neighboring pixels so that several inks produce in-between shades. Here: Floyd-Steinberg.

Index separation — a separation in which the artwork has a finite number of colors and the halftones come from the density of irregular dots rather than from the size of a halftone dot. It removes moiré between inks, but not the interference with the mesh — see chapter 1.

Opacity ceiling — the limit above which an ink cannot hit its target, because no such ink exists. Marked with a red `!`.

Perceived opacity — how covering an ink *looks*, on a scale where 70% means "seven tenths of the way to white". The slider in section 4. Photoshop wants a different number — see the table in chapter 9.

Registration mark — a mark for aligning the screens. A circle with a cross, identical in every channel.

Serpentine — reading every second row from the right, so that the error does not always drift the same way.

Spot channel — a Photoshop channel corresponding to one ink. The film is made from it.

Texture — how much the fabric eats the color of the ink. A preview parameter in the comparison window.

Underbase — the white printed under the colors, so that they come out on dark fabric.

ΔE — a measure of color difference. The smaller, the closer. Computed here as ΔE2000.

.dfpal / .dfink — application files: a saved palette and a backup of the ink database. Plain JSON inside.