

# The black box that isn't

Redaction fails silently, in five distinct ways — and every one of them is testable in about a minute

*For the person who signs off a document before it leaves the building | 08 September 2026*

A redaction failure does not announce itself. The document looks finished — black bars in all the right places — and the words underneath are still selectable, still searchable, still sitting in the file that has now been filed, served or published.

This paper sets out the five distinct ways the operation fails, what removal has to mean at the level of the file, and why the only trustworthy redaction is one the tool re-reads and can fail. It ends with a test you can run on whatever you use today.

## 1. The document that looks handled

The story is always the same, and it has run against courts, regulators, government departments and large law firms. A document is released with names or figures blacked out. Within hours somebody drags a cursor across a bar, presses copy, and pastes the text that was supposed to be gone.

Nothing malfunctioned. The tool was asked to draw a rectangle and it drew one. A rectangle is a drawing instruction; it sits above the words on the page in the same way a sticker sits above the print in a book, and removing it is about as difficult.

What makes this failure worse than doing nothing is the appearance of completion. An unredacted document gets checked, because everybody can see it is unredacted. A document with black bars across it has visibly been dealt with, so it moves down the chain without a second reader — and the second reader was the only remaining control.

*The risk here is not carelessness. It is a tool that reports success for an operation it did not perform, to an operator with no way of telling the difference.*

## 2. Five ways it fails, and how each one is found

These are distinct failures with distinct causes. A tool can get one right and all four others wrong, which is why a single satisfying test result proves less than it appears to.

| THE FAILURE          | WHAT IS ACTUALLY IN THE FILE                                                                                                                                                          | HOW IT COMES OUT                                                                                                              |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| The drawn rectangle  | Every character is intact underneath a filled shape. Common in general viewers, annotation tools and anything where the black bar is a comment, a highlight or a shape.               | Select across the bar and copy. Or search the document for a word you believe you removed.                                    |
| The removed layer    | The bar is an annotation object rather than page content, and annotations can be deleted, hidden or simply not drawn by another reader.                                               | Open the file in a different application from the one that wrote it. The bars may not be there at all.                        |
| The flattened page   | The text layer is gone, so the copy test passes. But flattening converts the whole page to a picture, so every other word on it stops being searchable too, and the file grows.       | Search for a word you did <i>not</i> redact. If nothing on the page is findable, the page was flattened rather than redacted. |
| The surviving pixels | The text was handled, but the value also appears in a scan, a photograph or a signature image on the same page. A shape over an image leaves the image untouched inside the file.     | Extract the images from the document and look at them on their own.                                                           |
| The rest of the file | The page is clean and the value survives in the document's own description of itself — the title, the author, prior filenames, an outline entry, an attachment, the revision history. | Read the document properties, then the raw file.                                                                              |

The third row deserves a note, because flattening is often recommended as the safe option. It does remove the text, and it also destroys the searchability of every unrelated word on the page, inflates the file, and defeats anyone who later needs to find a clause in a two-hundred-page bundle. It trades a real problem for a permanent one.

### 3. What removal has to mean

A page in a PDF is a program. It is a sequence of instructions — set a font, move the pen here, show this string, move there, show that one — and the characters you can read are arguments to a small set of text-showing operators: Tj, TJ, ' and ". Everything visible on the page arrives through one of them.

Redaction, done properly, is an edit to that program. The operators that draw the value are deleted from the content stream and the stream is written back without them. After that there is nothing to select, because there is no instruction left that would ever have drawn those characters.

The rectangle still gets painted — but afterwards, and for a different reason. It is the visible evidence that something was taken out, so that a reader can see the document has been edited rather than quietly abridged. **If the mark were doing the work, the work would not have been done.**

#### Why one pass is not always enough

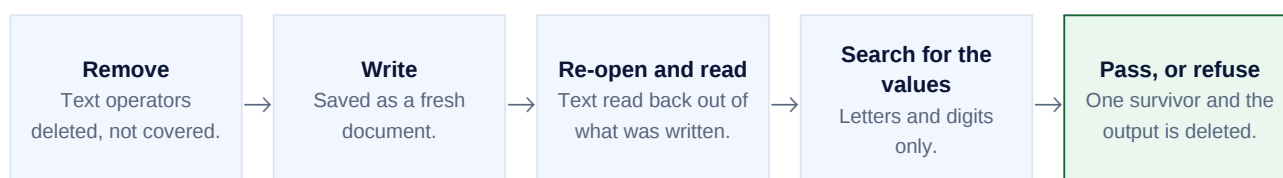
Documents do not lay text down the way a reader imagines. Sometimes a value sits in its own run, in its own table cell, and removing the run that begins inside the marked rectangle takes the number and leaves the label beside it — which is the tidy outcome.

Just as often the page draws Aadhaar 2345 6789 0124 as a single run beginning at the label, well outside the rectangle. Nothing starts inside the marked area, so a run-level removal removes nothing at all, and a tool that stops there has produced a rectangle over intact text — failure one, arrived at by a more sophisticated route.

The answer is to try the narrow removal, check the result, and if the value survives, widen to every run on that baseline that could reach the rectangle. That loses the label along with the value, which costs a little of the page and never leaves the number behind. Both attempts are verified; the narrow one is used only when it can be shown to have worked.

## 4. The check that separates redaction from theatre

Everything above is design. This is the part that makes it a claim rather than an intention: after the redacted file is written, it is opened again as a fresh document, its text is read out, and the values that were supposed to disappear are searched for.



*A tool with no failure state has no verification, because verification is the thing that produces one.*

The comparison is done on letters and digits only, with spacing and punctuation discarded on both sides. A reader may return a number with different spacing than the document used, and 2345 6789 0124 hiding as 234567890124 is still the number. A verification that can be defeated by a space is not a verification.

If any value survives, the file is **not offered as redacted**. The result is reported as failed, and the output that was written is removed from disk rather than left in the output folder where somebody would eventually find it and assume it was finished.

*This is the single question worth asking any vendor about this feature: can your redaction fail? A tool with no failure state has no verification, because verification is precisely the thing that produces one.*

### The masked variant, verified on the whole value

Indian practice often calls for masking rather than removal — UIDAI permits most entities to retain only the last four digits of an Aadhaar. Done badly this is the drawn rectangle again, with a helpful XXXX XXXX 1234 printed on top of the twelve digits that are still underneath.

Done properly the original run is removed like any other redaction, the masked form is drawn in its place as new text, and the verification then runs against **all twelve digits**. Keeping the last four is permitted; leaving the whole number under a label that says it was masked is the failure this feature exists to prevent.

## 5. Scans, photographs and signatures

The text path handles text. A great many documents in professional work are not text — they are a photograph of a statement, a scanned identity card, a signature block that is an image sitting on an otherwise ordinary page.

### A page with no text layer is refused

Where a redaction depends on detecting values automatically, a page that contains no text offers nothing to detect. Reporting such a page as scanned and clean would be the most dangerous possible answer, so it is refused outright and routed to recognition first. A rectangle over a scan is the false comfort this whole discipline exists to remove.

### Where a mark covers an image, the pixels go

The image is decoded out of the file, the covered region is painted into the pixel data, and the image stream is replaced with the edited one. Pulling the picture back out of the finished document yields the black area, because that is what the picture now contains.

For standalone images — a photograph of a document, marked up on its own — the same discipline runs one step further: the marks are burned in before the file is written, and the saved file is then read back and the marked region checked to be uniform. As with text, a redaction that cannot be verified is not reported as one.

### The placement that is refused rather than attempted

An image that has been rotated or skewed on the page is reported and skipped, not blanked. Mapping a rectangle through a rotation into pixel space is perfectly possible; getting it subtly wrong paints the wrong area and returns a file that looks redacted and is not. **A redaction that misses by a centimetre is a redaction that failed**, and the honest response to a case that cannot be done reliably is to say so and hand it back.

## 6. What cannot be proved, and is therefore not claimed

Verification works by searching the output for a known value. That mechanism has a boundary, and it is worth being explicit about where it falls.

- **A mark drawn by hand cannot be verified the same way.** When an operator draws over a signature, a photograph or a name no detector found, there is no value to search the output for. Its removal is reported rather than proved — and the result distinguishes the two, so a reviewer knows which marks carry evidence behind them and which carry only the operation.
- **Detection is rules, not judgement.** Identity and tax numbers are validated by check digit rather than matched by shape, so an invoice total is not reported as an identity number and a real one is not missed. But a rule finds what it describes. A name, a place or a fact that identifies somebody in context is the reviewer's job, and the scan is there to make that review shorter rather than to replace it.
- **Redaction here is a PDF operation.** It works by editing a content stream. A word processor file has no content stream to edit; convert it first, or do that work elsewhere.
- **This operation never modifies the source.** Redacted copies are written beside the original. That is deliberate — the original is often evidence — and it means the unredacted document still exists and is still your responsibility to handle.

### The metadata goes in the same pass

A document being redacted is, by definition, a document about to be circulated. Taking the identifiers off the page while leaving the author, the producing application, the original title, the revision history and any embedded scripting in place is a hole with no compensating benefit, so those can be stripped in the same operation.

Each file also produces a record: what was done, when, on which machine, with the SHA-256 of the input and of the output, how many values were removed, how many were masked, and which strategy the removal needed. That record is what lets somebody establish, months later, that this output came from that input.

## 7. Test the tool you already use

None of this needs to be taken on trust, including from this paper. Take a document, redact a value you know, and run these five checks against the output. They take about a minute and they map one-to-one onto the five failures in section 2.

- **Select and copy across the bar.** Then paste it somewhere. Anything that appears was never removed.
- **Search the finished file for the value.** Search without spaces as well as with them, because the two are different searches and only one of them is usually tried.
- **Search for a word you did not redact, on the same page.** No result means the page was flattened into a picture, and you have traded searchability for the appearance of safety.
- **Open it in a different application** from the one that produced it — a browser is enough. Bars that were annotations may simply not be drawn.
- **Read the document properties.** Author, title, producing software, and any earlier name the file carried.

If the tool passes all five, it is doing the work. If it fails any of them, the failure is silent, has been silent on every document it has processed so far, and will stay silent until somebody outside the firm finds it.

*Run this against a document that has already gone out, not only against a fresh test file. The interesting question is not whether the tool can do it correctly — it is what is already in the hands of the people you sent it to.*

## 8. Conclusion

Redaction is the one document operation where a silent partial success is worse than a visible refusal. Everywhere else, a tool that does eighty per cent of the job leaves you with twenty per cent to finish. Here it leaves you with a document that has been signed off, distributed, and is carrying exactly the material it was supposed to have lost.

So the standard is not that the tool is careful. It is that the tool checks its own work against the finished file, tells you plainly when it could not do the job, and refuses to hand you the output when the check comes back wrong — including in the cases, honestly marked, where no check is possible.

Everything described here is verifiable on a document you control, with software you already have. That is the only kind of security claim worth making, and the five checks above are how you make somebody else's.