

# PdfCraft

ALPHA

clone of Adobe Acrobat Pro

One headless MCP server, pdfcraft: open, organize, OCR, redact, fill, sign, compare and check PDFs. You describe the job; Claude finds the right tool and shows you the page.

**v0.8.0**

Driven from Claude Code via MCP  
Richard Young • richard@deepneuro.ai  
2026-10-15

**Alpha software.** PdfCraft is a pre-1.0 ArtCraft app. Expect missing features, rough edges and the occasional crash; keep originals safe and let Claude work on copies.

**163**

MCP tools in total

**17**

listed in compact mode

**194**

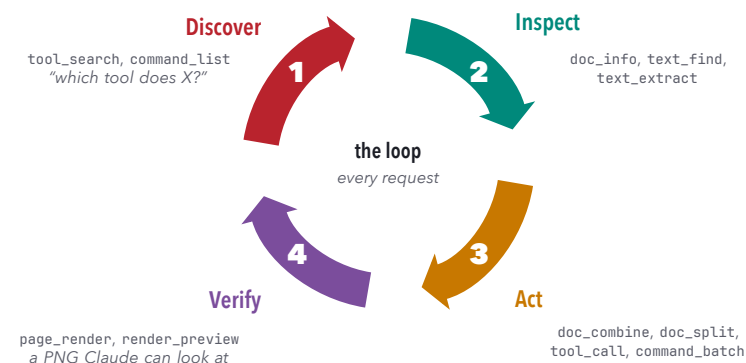
app commands

**64**

automatable by id

## 01 The idea in one picture

You never open the PdfCraft window. You type a goal in Claude Code; Claude opens the PDF through the pdfcraft server, edits it in memory (every step undoable), renders a page so it can see what happened, and saves only when you ask.



**i Check it is wired up**

In Claude Code type `/mcp: pdfcraft` should show *connected* with 17 tools. There is **no** pdfcraft-live: the server runs the engine headless over stdio, opens no network port, and never touches the PdfCraft window (open or not). Nothing hits disk until `doc_save`, so always finish with “save it as ...”.

**✓ Compact mode: how Claude reaches all 163 tools**

`tool_search` query / category → `tool_search` name (full schema) → `tool_call` name + arguments.

Naming the tool in your prompt (“use `doc_audit_redactions`”) skips the search. Tools marked <sup>†</sup> in this guide go through `tool_call`. Alternative route: `command_run` / `command_batch` with menu command ids such as `page.rotate` or `ocr.recognize` (194 registered, 64 automatable).

**! The file root**

The server is started with `--root` set to your home folder. A path outside it is refused with “...is outside the allowed directory”: copy PDFs into e.g. `~/pdf-work/` before asking. Outputs (`out`, `out_dir`, CSVs, PNGs, .p12 IDs) must land under the root too.

## 02 First five minutes

Every image on this page is real output from PdfCraft, captured through the MCP server while writing this guide. Paste the prompts as they are.

- 1 Put a PDF where the server can see it** Copy any PDF to `~/pdf-work/` (anything under your home folder works).
- 2 Open and look** “Use pdfcraft: open `~/pdf-work/report.pdf`, tell me how many pages, and render page 1.” (storyboard 1).
- 3 Change something** A watermark, a rotation, a page move. Each is one undoable step (storyboard 2).
- 4 Look again** “Render page 2 so I can see it.” Claude compares before and after and tells you what changed.
- 5 Save to a new name** “Save as `~/pdf-work/report-draft.pdf`.” The original is untouched.

## Storyboard 1: open a PDF and see a page

### YOU, IN CLAUDE CODE

"Use pdfcraft. Open ~/craft-guides-demo/sample-report.pdf, tell me how many pages it has and whether it is editable, then render page 1."

### MCP CALLS CLAUDE MADE

- 1 **doc\_open** {"path": "~/craft-guides-demo/sample-report.pdf" → {"doc": 1, "pages": 5, "editable": true, "encrypted": false}
- 2 **page\_render** {"doc": 1, "page": 1, "dpi": 80}

### WHAT CAME BACK

WordCraft

Driving it from Claude Code via MCP

1

WordCraft

ALPHA

v0.7.0

clone of Microsoft Word

Drive a Word-compatible word processor from Claude Code: write, restyle, cite, redline and mail-merge .docx.

Study guide • 2026-10-14 • prepared for Ric

Alpha software. WordCraft is a pre-1.0 ArtCraft app. Expect missing features, rough edges and the occasional crash; keep originals safe and let Claude work on copies.

16

MCP tools

617

engine commands

1

MCP server

v0.7.0

version

### The mental model

- Everything is a command id. Ribbon actions are ids like `format.bold` or `references.toc`. Claude finds them with `list_commands` (query: "toc", "merge"... ) and runs them with `execute` or `batch`.
- The loop: open/create → select (select\_text) → run ids → verify (inspect\_document, render\_page) → save\_document.
- Save format = file extension: docx, pdf, odt, rtf, html, md, tex, txt, png.

### How it connects

2

Claude Code

you type a goal

→ MCP / stdio

wordcraft

wordcraft-cli mcp

headless, in memory

→

.docx / .pdf / .md

files on disk

1 Check it

Type /mcp in Claude Code: wordcraft should show connected. Smoke test: "Use wordcraft to open the sample document and tell me how many pages it has." There is **no live-window server** registered for WordCraft, so the window tools (`screenshot`, `click`, `key`, `type`) won't work.

- 1 Page 1 at 80 dpi: the banner of the WordCraft guide used as a sample document.
- 2 Live text, not an image: `text_find` and `text_extract` work on it. The first attempt used a path on an external volume and was refused ("is outside the allowed directory"); Claude copied the file under the home folder and tried again.

## Storyboard 2: a DRAFT watermark through tool\_call

### YOU, IN CLAUDE CODE

"Put a diagonal DRAFT watermark at 25% opacity on every page, then render page 2 so I can check it."

### MCP CALLS CLAUDE MADE

- 1 **tool\_search** {"query": "watermark"} → finds doc\_watermark (not in the compact core set)
- 2 **tool\_call** {"name": "doc\_watermark", "arguments": {"doc": 1, "text": "DRAFT", "opacity": 0.25, "rotation": 45}} → {"dirty": true, "undo": "Add watermark"}
- 3 **page\_render** {"doc": 1, "page": 2, "dpi": 80}

### WHAT CAME BACK

WordCraft Driving it from Claude Code via MCP

## Goals → how to ask

### 1 Create, convert & automate

#### 1 Draft a report, save DOCX + PDF

ASK CLAUDE  
"Start a report with a title, Introduction and Methods from these notes; save as ~/Documents/report.docx and a PDF."

TOOLS **new\_document** → **type\_text** → **execute** → **save\_document**

KEY template: report; para.heading1; para.style; file.exportPdf

#### 2 Convert between formats

ASK CLAUDE  
"Convert ~/thesis/ch3.md to Word, and proposal.docx to ODT and PDF."

TOOLS **open\_document** → **save\_document**

KEY reads docx, doc, odt, rtf, md, tex, html, txt; writes by extension

• Bulk: wordcraft-cli convert in out in a loop.

#### 3 Use a template's styles

ASK CLAUDE  
"Create a memo from ~/templates/memo.dotx, or copy its Heading styles into this report."

TOOLS **execute**

KEY file.newFromTemplate; styles.copyFrom (path, styles); tools.attachTemplate

#### 4 Record a reusable macro

ASK CLAUDE  
"Define a macro 'course-format' (Georgia headings, 1-inch margins) and run it on each syllabus."

TOOLS **execute** → **save\_document**

KEY tools.macros (define: name + steps, run)

• Persistence across sessions not documented; a replayed batch works too.

### 2 Edit & format

#### 5 Find and replace (regex too)

ASK CLAUDE  
"Replace every 'Licensee' with 'Customer' (whole word) and reformat dates like 3/7/26 as 2026-03-07."

TOOLS **execute** → **get\_text**

KEY find (regex); replace (text, with, global)

#### 6 Restyle the whole document

ASK CLAUDE  
"Give this a cleaner look: new theme and style set, Georgia headings, open paragraph spacing."

TOOLS **batch** → **render\_page**

KEY batch (steps); render\_page (page)

- 1 The diagonal DRAFT watermark at 25% opacity, applied to every page by one **tool\_call**.
- 2 The page content is untouched underneath; the reply carried undo: "Add watermark", so "undo that" is one **edit\_undo** call.

**i A gotcha Claude hit, and fixed, while making this page**

`text_find` takes `query`, not `text`. The first call returned “unknown argument “text” (expected: doc, limit, query, skip, start\_page)”; Claude read the message and re-sent it. Error replies name the real parameters, which is why “try it and read the error” is a fine strategy here.

**Prompts to explore with**

●●● warm-up   ●●● real work   ●●● stretch

●●●

“Use pdfcraft: search the tools in the “form” category and list what each one does.”

`tool_search` by `category`: see the 163-tool surface without reading a manual.

●●●

“Open ~/pdf-work/any.pdf and give me its metadata, bookmarks, fonts and any form fields.”

`doc_info` reads everything; nothing is changed.

●●●

“Find every “indemnify” in the contract, tell me the pages, and render the first one.”

`text_find` `query` + `render_preview`.

●●●

“Combine cover.pdf, pages 1-3 of report.pdf and scan.jpg into packet.pdf.”

`doc_combine` is a core tool: one call.

●●●

“Rotate the landscape pages, delete page 7, move page 12 to the front, save as a new file.”

Three `tool_calls` and a save; Claude re-reads `doc_info` between steps.

●●●

“Audit release.pdf for fake redactions, then redact all SSNs and emails properly, save a copy and re-audit it.”

`doc_audit_redactions`<sup>†</sup>, `redact_mark`<sup>†</sup>, `redact_apply`<sup>†</sup>, full-rewrite save.

●●●

“OCR every scan in ~/pdf-work/inbox, make them PDF/A-2b, and give me a table of pages and words per file.”

Stretch: `doc_batch`<sup>†</sup> or a loop of `ocr_recognize`<sup>†</sup> + `pdfa_convert`<sup>†</sup>.

## 03 Goals ? how

Each card is a prompt you can paste, the tools Claude reaches for, and the parameters doing the work. Pages are 1-based; geometry is in points with the origin top-left. Open docs are also resources: pdfcraft://doc/{doc}/info, /text, /page/{page}/image.

### ■ Organize pages

#### 1 Merge PDFs and images into one packet

##### ASK CLAUDE

"Combine cover.pdf, pages 1-3 of report.pdf and scan.jpg into packet.pdf."

TOOLS `doc_combine`

KEY `paths` (in order), `pages` per path ("1-3" or null), `out`, `field_conflicts` rename|merge. Images and .txt are converted; one bookmark per file.

#### 2 Split a long PDF into chapters

##### ASK CLAUDE

"Split thesis.pdf at its top-level bookmarks into ~/pdf-work/chapters/."

TOOLS `doc_open` → `doc_split`

KEY `out_dir` + one of `every`, `before`, `bookmarks`, `max_mb`. Hand-picked pages: `page_extract`<sup>†</sup> (`pages`, `out`, `separate`).

#### 3 Rotate, delete, move, insert pages

##### ASK CLAUDE

"Rotate the even landscape pages 90 degrees, delete page 7, move page 12 to the front, save as scan-fixed.pdf."

TOOLS `page_rotate`<sup>†</sup> → `page_delete`<sup>†</sup> → `page_move`<sup>†</sup> → `doc_save`

KEY `degrees` (+ = clockwise), `subset` even|odd, `orientation`; `page_move` `pages`, `to`. Also `page_insert_file`<sup>†</sup>, `page_duplicate`<sup>†</sup>.

#### 4 Page numbers, Bates, labels, watermark

##### ASK CLAUDE

"Footer 'Page 1 of n', label front matter i-iv, and a 30% DRAFT watermark on every page."

TOOLS `doc_header_footer`<sup>†</sup> → `page_number`<sup>†</sup> → `doc_watermark`<sup>†</sup>

KEY `footer_center` with tokens like <<Page 1 of n>> or Bates <<Bates Number#6#1#PREFIX#SUFFIX>>; `style` lower-roman; `text`, `opacity`, `rotation` (storyboard 2).

## Scan, OCR & extract

### 5 Make a scanned PDF searchable

#### ASK CLAUDE

"OCR lease-scan.pdf at 300 dpi, save as lease-searchable.pdf, and report words per page."

**TOOLS** `ocr_status`<sup>†</sup> → `ocr_recognize`<sup>†</sup> → `doc_save`

**KEY** `dpi` 72-600 (default 300), `pages`, `language` (only en), `skip_text_pages`. Adds invisible text; image unchanged.

- Ask for `ocr_status` first: it says whether the OCR models are installed.

### 8 Pull text and tables for analysis

#### ASK CLAUDE

"Extract the table in box (50,120)-(560,700) on pages 5-9 as tab-separated rows; dump all text to 10-K.txt."

**TOOLS** `text_find` → `text_extract` → `doc_export_text`<sup>†</sup>

**KEY** `text_extract rect` = column select (one row per line, cells tab-separated), `pages`.

- To Office: `doc_export_office`<sup>†</sup> (.docx/.pptx/.html/.rtf; layout is lossy).

### 6 Batch-OCR and shrink a folder

#### ASK CLAUDE

"Convert, OCR and optimize everything in ~/pdf-work/inbox (recursively) into ~/pdf-work/out."

**TOOLS** `doc_batch`<sup>†</sup>

**KEY** `paths`, `out_dir`, `steps` ["convert", "ocr", "optimize"] (convert first), `recursive`, `ppi`, `quality`. Never overwrites; sources untouched.

- Also `ocr_recognize_files`<sup>†</sup>, or `action_run`<sup>†</sup> "Optimize Scanned Documents".

### 7 Straighten a phone photo of a page

#### ASK CLAUDE

"Page 1 of receipt.pdf is a skewed photo: flatten it to clean black-and-white, then OCR it."

**TOOLS** `page_enhance_camera`<sup>†</sup> → `ocr_recognize`<sup>†</sup>

**KEY** `corners` (8 numbers, omit = auto-detect), `mode` colour|greyscale|bw, `level` 0-100 (60+ removes specks).

## Redaction & security

### 9 Find fake redactions

#### ASK CLAUDE

"Audit release.pdf for black boxes over live text and list each hidden string with its page."

**TOOLS** `doc_open` → `doc_audit_redactions`<sup>†</sup>

**KEY** Finds covered-text (dark box painted over extractable text) and unapplied-mark (/Redact never applied). Fails closed on unreadable content.

● Shell: `pdfcraft-cli audit f.pdf` exits non-zero on findings.

### 10 Redact properly: SSNs, emails, names

#### ASK CLAUDE

"Redact all SSNs, emails and 'J. Smith' in case.pdf with FOIA code (b)(6); apply, save a copy, re-audit."

**TOOLS** `redact_mark`<sup>†</sup> → `redact_apply`<sup>†</sup> → `doc_save`

**KEY** `pattern` ssn|email|phone|credit-card|date, words, find, rect; `code_set` foia|privacy-act + codes, or `over`lay. Apply runs a verification pass.

### 11 Strip hidden information

#### ASK CLAUDE

"Show what hidden data board-deck.pdf carries, then remove metadata, comments and scripts."

**TOOLS** `doc_hidden_info`<sup>†</sup> → `doc_remove_hidden`<sup>†</sup> → `doc_save`

**KEY** `categories`: metadata, comments, attachments, hidden-text, links-actions-scripts, ... (none = full Sanitize). Forms get flattened.

● One shot: `action_run`<sup>†</sup> "Prepare for Distribution".

### 12 Password-protect for sharing

#### ASK CLAUDE

"Encrypt board-deck.pdf with AES-256, open password q3-board, no copying, low-res printing only."

**TOOLS** `doc_protect`<sup>†</sup> → `doc_save`

**KEY** `open_password`, `permissions_password` (needed for any restriction), `printing` none|low|high, `copy`, `changes`, `compatibility`.

● Undo with `doc_unprotect`<sup>†</sup>.

## Forms & signatures

### 13 Fill a form from your data

#### ASK CLAUDE

"List the fields in w9.pdf, fill them from this table, tick LLC, flatten, save w9-filled.pdf."

TOOLS `form_fields`<sup>†</sup> → `form_fill`<sup>†</sup> → `doc_flatten`<sup>†</sup> → `doc_save`

KEY values: name → string, true/false, or string list. One undo step. Bulk: `doc_import_data`<sup>†</sup> (XFDF/FDF/XML/CSV).

### 16 Sign digitally and verify

#### ASK CLAUDE

"Create a self-signed ID for me, sign contract.pdf on page 3 (reason: Approved), then verify it."

TOOLS `sign_id_create`<sup>†</sup> → `sign_document`<sup>†</sup> → `sign_list`<sup>†</sup>

KEY id (.p12), password, out, page+rect or field, reason, certify. PAdES B-B, SHA-256.

● No fields? `fill_sign_add`<sup>†</sup> types text, checks, dates, signatures.

### 14 Make a flat form fillable

#### ASK CLAUDE

"Detect the blanks and checkboxes in intake.pdf, show me the proposals, then add them."

TOOLS `form_detect_fields`<sup>†</sup> → `form_add_field`<sup>†</sup> → `doc_save`

KEY add false = propose only; then true. Tidy with `form_set_props`<sup>†</sup>, `form_tab_order`<sup>†</sup>.

● Detected rects use a bottom-left origin.

### 15 Collect responses into a CSV

#### ASK CLAUDE

"Merge field values from every PDF in ~/pdf-work/responses into responses.csv."

TOOLS `form_merge_data`<sup>†</sup>

KEY paths (PDF/FDF/XFDF), path (.csv): one column per field, one row per file. Single file: `doc_export_data`<sup>†</sup> what fields.

## Review, compare & publish

### 17 Compare two contract versions

#### ASK CLAUDE

"Compare v2 against v1, summarise the wording changes, write a report PDF, and mark changes in v2."

TOOLS `doc_compare`<sup>†</sup> → `doc_compare_report`<sup>†</sup> → `doc_compare_mark`<sup>†</sup>

KEY doc = newer, other = older; visual also diffs appearance. Marks: blue replaced, green inserted, red deleted.

### 18 Annotate and summarise comments

#### ASK CLAUDE

"Highlight every 'indemnify' on page 4, stamp page 1 DRAFT, then a comment summary by author."

TOOLS `comment_add`<sup>†</sup> → `comments_summarize`<sup>†</sup>

KEY type (highlight, note, stamp, textbox, ...), find+all, stamp draft; summary sort author, out.

### 19 Accessibility check and fixes

#### ASK CLAUDE

"Check handout.pdf for accessibility, fix title and language (en-US), list figures lacking alt text."

TOOLS `accessibility_check`<sup>†</sup> → `accessibility_fix`<sup>†</sup> → `accessibility_set_alt`<sup>†</sup>

KEY 32 rules, 7 categories; fix rule title|primary-language|tab-order + value. HTML report: `accessibility_report`<sup>†</sup>.

**20 Archive as PDF/A and shrink****ASK CLAUDE**

"Verify paper.pdf against PDF/A-2b, convert it, then write a smaller copy and say where the bytes went."

**TOOLS** `pdfa_verify`<sup>†</sup> → `pdfa_convert`<sup>†</sup> → `doc_reduce`<sup>†</sup>

**KEY** `level 2b|3b`. `doc_audit_space`<sup>†</sup> shows bytes by kind; `doc_optimize`<sup>†</sup> for custom `ppi`/JPEG settings.

**✓ Always add: "render it and check"**

`page_render` (`dpi` up to 600) and `render_preview` (`max_side`) return PNGs Claude can actually see. Ask for a render after edits; Claude will compare it with the earlier one and tell you what changed before you save.

# 04 The command map (194 app commands)

Every menu item is a dotted command id; 64 of them can run through `command_run` and are mapped to a tool. Darker tiles are bigger families. Ask Claude to "list pdfcraft commands matching *word*" or "search pdfcraft tools in category *X*" to open any tile.

|                                                                                             |                                                                                      |                                                                                                          |                                                                                                         |                                                                          |                                                                                 |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>26</b> <b>comment.</b><br>add, reply, status, mark, lock, hide, summarize, image preview | <b>22</b> <b>edit.</b><br>undo, redo, text edit, content, objects, images            | <b>21</b> <b>form.</b><br>fields, fill, reset, add/delete, props, tab order, detect, merge data, scripts | <b>18</b> <b>page.</b><br>rotate, delete, move, insert, extract, replace, boxes, number, enhance camera | <b>18</b> <b>view.</b><br>zoom, layout, reading colors, panels (UI only) | <b>17</b> <b>sign.</b><br>IDs, sign, verify, trust, keychain and Windows stores |
| <b>10</b> <b>file.</b><br>open, save, save as, close, combine, split, print                 | <b>7</b> <b>measure.</b><br>distance, perimeter, area, scale, snap, export           | <b>7</b> <b>redact.</b><br>mark by pattern/words/rect, apply, clear, audit                               | <b>7</b> <b>export.</b><br>text, images, office, data, revisions                                        | <b>7</b> <b>help.</b><br>about, docs, shortcuts (UI only)                | <b>6</b> <b>create.</b><br>blank PDF, from files, from text, multiple           |
| <b>6</b> <b>ally.</b><br>check, report, fix, figures, alt text, tags                        | <b>5</b> <b>protect.</b><br>protect, unprotect, hidden info, remove hidden, sanitize | <b>6</b> <b>ocr. / actions.</b><br>status, recognize, recognize files; action wizard list, run           | <b>11</b> <b>other</b><br>optimize, tools, bookmark, print, standards (PDF/A), ui                       |                                                                          |                                                                                 |

# 05 Tool reference

## Core set (visible in compact mode)

| MCP tool       | What it does                                                                                        |
|----------------|-----------------------------------------------------------------------------------------------------|
| doc_open       | Open a PDF ( <code>path</code> , <code>password</code> ); returns doc id and page count.            |
| doc_info       | Metadata, pages, bookmarks, annotations, fields, links, fonts, security.                            |
| doc_inspect    | <code>doc_info</code> for one doc, or every open doc.                                               |
| doc_save       | In place (incremental) or to <code>path</code> (full rewrite); atomic.                              |
| doc_close      | Close; refuses unsaved changes unless <code>discard_changes</code> .                                |
| page_render    | One page to PNG ( <code>dpi</code> default 96, max 600).                                            |
| render_preview | Bounded PNG preview ( <code>max_side</code> , default 1024).                                        |
| text_extract   | Reading-order text; <code>rect</code> for column/table select.                                      |
| text_find      | Phrase search ( <code>query</code> ) with page and line rectangles.                                 |
| doc_combine    | Merge PDFs, images, .txt in order into one file.                                                    |
| doc_split      | Split by <code>every</code> , <code>before</code> , <code>bookmarks</code> or <code>max_mb</code> . |
| edit_undo      | Undo the last edit (redo: <code>edit_redo</code> <sup>†</sup> ).                                    |
| command_list   | All 194 app commands with menu, shortcut and mapped tool.                                           |
| command_run    | Run one command id with <code>params</code> .                                                       |
| command_batch  | Up to 1000 { <code>id</code> , <code>params</code> } steps; <code>stop_on_error</code> .            |
| tool_search    | Find tools by <code>query/category</code> ; <code>name</code> gives the schema.                     |
| tool_call      | Run any tool: <code>name</code> + <code>arguments</code> .                                          |

## Everything else, by category (via tool\_call)

| Category               | Tools                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| doc                    | <code>doc_batch</code> , <code>doc_list</code> , <code>doc_set_info</code> , <code>doc_create</code> , <code>doc_create_multiple</code> , <code>doc_export_data</code> , <code>doc_import_data</code> , <code>doc_hidden_info</code> , <code>doc_remove_hidden</code> , <code>doc_audit_redactions</code> , <code>doc_print</code> , <code>doc_protect</code> , <code>doc_unprotect</code> , <code>doc_header_footer</code> , <code>doc_watermark</code> , <code>doc_background</code> , <code>doc_remove_marks</code> , <code>doc_export_images</code> , <code>doc_export_all_images</code> , <code>doc_export_text</code> , <code>doc_export_office</code> , <code>doc_compare</code> , <code>doc_compare_report</code> , <code>doc_compare_mark</code> , <code>doc_reduce</code> , <code>doc_optimize</code> , <code>doc_audit_space</code> , <code>doc_initial_view</code> , <code>doc_revisions</code> , <code>doc_open_revision</code> , <code>doc_flatten</code> |
| page                   | <code>page_rotate</code> , <code>page_delete</code> , <code>page_move</code> , <code>page_insert_blank</code> , <code>page_insert_file</code> , <code>page_extract</code> , <code>page_replace</code> , <code>page_duplicate</code> , <code>page_set_box</code> , <code>page_number</code> , <code>page_add_text</code> , <code>page_add_image</code> , <code>page_images</code> , <code>page_enhance_camera</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| text                   | <code>text_lines</code> , <code>text_paragraphs</code> , <code>text_edit</code> , <code>text_edit_layout</code> , <code>text_fonts</code> , <code>text_font_files</code> , <code>text_font_import</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| content, object, image | <code>content_list</code> , <code>content_update</code> , <code>content_delete</code> , <code>object_list</code> , <code>object_move</code> , <code>image_hit</code> , <code>image_edit</code> , <code>image_save</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| bookmark               | <code>bookmark_list</code> , <code>bookmark_add</code> , <code>bookmark_rename</code> , <code>bookmark_delete</code> , <code>bookmark_move</code> , <code>bookmark_set_page</code> , <code>bookmark_from_structure</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Category           | Tools (cont.)                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| link(s)            | link_list, link_open, link_add, link_edit, link_delete, links_from_urls, links_remove                                                                                                                                                                                                                                                                            |
| form               | form_fields, form_fill, form_reset, form_add_field, form_delete_field, form_set_image, form_set_props, form_tab_order, form_detect_fields, form_merge_data, form_set_script, form_actions, form_set_actions                                                                                                                                                      |
| fill, stamp        | fill_sign_add, fill_sign_date_format, stamp_custom                                                                                                                                                                                                                                                                                                               |
| comment(s)         | comment_list, comment_add, comment_reply, comment_set_status, comment_mark, comment_lock, comment_edit, comment_delete, comment_image_preview, comments_hide, comments_summarize                                                                                                                                                                                 |
| redact             | redact_mark, redact_apply, redact_clear                                                                                                                                                                                                                                                                                                                          |
| sign               | sign_list, sign_id_create, sign_document, sign_trust, sign_keychain_ids (macOS), sign_windows_ids (Windows)                                                                                                                                                                                                                                                      |
| ocr                | ocr_status, ocr_recognize, ocr_recognize_files                                                                                                                                                                                                                                                                                                                   |
| accessibility      | accessibility_check, accessibility_report, accessibility_fix, accessibility_figures, accessibility_set_alt, accessibility_tags, accessibility_set_tag_property, accessibility_move_tag, accessibility_autotag_preview, accessibility_autotag_apply, accessibility_review_record, accessibility_review_save, accessibility_review_load, accessibility_review_html |
| measure            | measure_distance, measure_perimeter, measure_area, measure_info, measure_list, measure_scale, measure_snap, measure_export                                                                                                                                                                                                                                       |
| pdfa, print        | pdfa_verify, pdfa_convert, print_readiness, printers, printer_options                                                                                                                                                                                                                                                                                            |
| js                 | js_run, js_document_scripts, js_set_document_script, js_enabled                                                                                                                                                                                                                                                                                                  |
| action, view, edit | action_list, action_run, view_reading_colors, edit_redo                                                                                                                                                                                                                                                                                                          |

**i Action Wizard presets (action\_run)**

Built-in `action` names: "Prepare for Distribution", "Optimize Scanned Documents", "Add Page Numbers", "Mark as Confidential", "Sanitize Documents". Or pass custom `steps`: `recognize_text` `reduce_file_size` `remove_hidden_information` `sanitize` `flatten_comments` `flatten_fields` `add_watermark` `add_header` `add_footer` `set_title` `detect_form_fields` `run_javascript` (the watermark/header/footer/title/javascript steps take an `arg`).

## 06 Example tool calls

### Storyboard 2: open, watermark through tool\_call, render, save a copy

```
doc_open      {"path": "~/craft-guides-demo/sample-report.pdf"}    -> {"doc": 1, "pages": 5}
tool_call     {"name": "doc_watermark",
               "arguments": {"doc": 1, "text": "DRAFT", "opacity": 0.25, "rotation": 45}}
page_render   {"doc": 1, "page": 2, "dpi": 80}                    -> PNG
doc_save      {"doc": 1, "path": "~/craft-guides-demo/sample-report-draft.pdf"}
```

### Audit, redact, save a copy (non-core tools wrapped in tool\_call)

```
doc_open      {"path": "~/pdf-work/release.pdf"}                -> {"doc": 1, ...}
tool_call     {"name": "doc_audit_redactions", "arguments": {"doc": 1}}
tool_call     {"name": "redact_mark",
               "arguments": {"doc": 1, "pattern": "ssn",
                             "code_set": "foia", "codes": ["(b)(6)"]}}
tool_call     {"name": "redact_mark",
               "arguments": {"doc": 1, "words": ["J. Smith"], "whole_words": true}}
tool_call     {"name": "redact_apply", "arguments": {"doc": 1}}
doc_save      {"doc": 1, "path": "~/pdf-work/release-redacted.pdf"}
```

### Combine a cover, three report pages and a scan (core tool)

```
doc_combine {"paths": ["~/pdf-work/cover.pdf",
                       "~/pdf-work/report.pdf",
                       "~/pdf-work/scan.jpg"],
            "pages": [null, "1-3", null],
            "out":   "~/pdf-work/packet.pdf"}
```

### Rotate, delete, OCR and save in one command\_batch (menu command ids)

```
command_batch {"stop_on_error": true, "steps": [
  {"id": "page.rotate", "params": {"doc": 1, "degrees": 90, "subset": "even"}},
  {"id": "page.delete", "params": {"doc": 1, "pages": [7]}},
  {"id": "ocr.recognize", "params": {"doc": 1, "dpi": 300}},
  {"id": "file.save", "params": {"doc": 1, "path": "~/pdf-work/scan-fixed.pdf"}}]}
```

## 07 Limits & tips

### ! Saving in place keeps the old bytes

`doc_save` without `path` appends an incremental update, so earlier content stays inside the file. Great for signatures, wrong for redaction or sanitizing: always save those to a new path (full rewrite) and re-run `doc_audit_redactions`<sup>†</sup> on the result.

### ! Alpha: know the thin spots

- README (Oct 2026): about 55% of the way to replacing Acrobat for everyday work.
- Thin: editing existing text (`text_edit`<sup>†</sup>), OCR beyond English, Office export (layout lossy), Preflight, full XFA.
- Fuzzing still finds crashes on hostile files; work on copies, not originals.

### ✓ Habits that work

- Nothing hits disk until `doc_save` (tools with their own `out/path` excepted). Ask for new filenames.
- Ask Claude to `render_preview` after edits; it can look at the PNG and self-check.
- Page numbers shift after deletes/moves: have Claude re-read `doc_info` between steps.
- Error replies list the real parameter names (see the `text_find` note above); let Claude read them and retry.
- For scripts outside Claude: `pdfcraft-cli combine, split, audit, batch, run`.

### i Gotchas

- Coordinates are top-left origin, except `doc_compare`<sup>†</sup> and `form_detect_fields`<sup>†</sup> results (bottom-left).
- `doc_protect`<sup>†</sup> and `doc_remove_hidden`<sup>†</sup> force a full rewrite on the next save.
- Side effects: `doc_print`<sup>†</sup> can reach a real printer (ask for a print-ready `path` instead); `js_run`<sup>†</sup> runs document JavaScript; `sign_trust`<sup>†</sup> edits the trust list.