

Kuonen.PDFTools — Roadmap

Farb-Legende (Status):

 fertig	 teilweise	 offen	 neu / Baustelle	 entfaellt / nie
---	--	--	--	--

Stand: aktueller Code M1347. Git-versionierte Kontroll-Roadmap; das Download-PDF (/download/roadmap) wird ab M1188 direkt aus dieser Datei erzeugt. Alles steht als Tabelle, kaum Fliesstext.

Aufwand: S klein · M mittel · L gross · XL sehr gross.

Status: Fertig · Teilweise · Offen · Zurueckgestellt · Neu/Baustelle · x Nie im PDF.

Aufbau: v4 = geplante Werkzeug-Paritaet (Acrobat/InDesign/Word/Excel/PowerPoint/FOP/LaTeX/HTML/Dreamweaver/Notepad + Bildformate/Schriften/Editor) + externe Punkte · v3 = abgeschlossene grosse Tracks · v2 = History · v1 = Archiv je Bereich.

Verbindliche Wahrheit je Meilenstein: das Ledger in CLAUDE.md.

v4 · Werkzeug-Paritaet - geplant (offen)

Ehrliche Lueckenanalyse im Code geprueft (nicht aus dem Gedaechnis): was Editor/Bibliothek gegenueber den grossen Vorbildern noch WENIGER koennen. Alles baubar (Grundprinzip: keine „bauen wir nie“-Grenzen).

Bereits Gebautes ist bewusst NICHT gelistet (z. B. verschachtelte/GREP-Formate, Mehrspaltensatz, Silbentrennung,

Index/TOC/Querverweise/Fussnoten, Data Merge, IDML — schon da). Gruppen: AP Acrobat · LID InDesign ·

LWD Word · LXL Excel · LFO FOP · LTX LaTeX · LHT HTML · LPP PowerPoint ·

IMG Bildformate · FNT Schriften · TXT Notepad/Klartext · AG Apogee-Rest · ED Editor-Paritaet · PRN virtueller Drucker · LR Lightroom.

v4 auf einen Blick

Zwei verschiedene Zahlen — nicht verwechseln. „Punkte“ zählt nur die Restliste unten (v4 listet ausschliesslich Luecken; alles Gebaute steht in v1–v3). „Abdeckung“ ist die grobe Einschätzung, wie weit das Werkzeug insgesamt ersetzt ist — eine Gruppe kann 0 erledigte Punkte haben und trotzdem 85 % abdecken.

Grp	Vorbild	n	offen	Abdeckung (grob)	Groesste Luecke
AP	Acrobat Pro	7	4	~90 %	AP2 Feld-Autoerkennung, AP3 Bild in-place, AP7 Editor in-place
LID	InDesign	6	1	~95 %	nur noch die hilfslinienbasierte Liquid-Layout-Regel (LID6)
LWD	Word	6	2	~88 %	LWD6 Textrahmen/Formen/SmartArt, OMML schreiben
LXL	Excel	6	0	~90 %	Diagramme/Bilder schreiben, Pivot
LFO	Apache FOP	5	0	~98 %	nur noch PCL/AFP (nicht abbildbar)
LTX	LaTeX	6	4	~90 %	LTX1 TikZ, LTX2 BibTeX
LHT	HTML/CSS	6	0	~95 %	nur Medium-Grenzen (Events/Timer im statischen PDF)
LPP	PowerPoint	9	9	~45 %	alles ausser positionierten Text-/Bildrahmen
IMG	Bildformate	13	13	Lesen ~95 % · Schreiben ~35 %	nur PNG/JPEG/1-Bit-TIFF/JBIG2 schreibbar
FNT	Schriften	7	1	Satz ~98 % · Datei-Ausgabe ~90 %	nur noch der volle Hinting-Befehlssatz (FNT4)
TXT	Notepad	6	0	~98 %	komplett; C#-Hervorhebung ohne Format-Angabe im Interpolationsausdruck
AG	Apogee	4	0	~98 %	komplett
ED	Editor (alle Vorbilder)	24	24	InDesign ~80 % · Acrobat ~30 % · Word ~35 % · Excel ~5 % · PowerPoint 0 % · Dreamweaver ~10 %	Dienste sind da, die Oberflaeche fehlt
LR	Lightroom	14	14	~10 % (Farb-/Bild-Unterbau da)	Entwicklung und Katalog fehlen
PRN	virtueller Drucker	8	6	~35 %	der IPP-Codec (M1327) und der Server stehen; es fehlen Windows-/macOS-Treiber, Bekanntmachung im Netz und die Warteschlangen-Oberflaeche
EXT	extern	4	—	—	toolchain-/zertifikatsgebunden, kein fehlender Code

Dreamweaver hat keine eigene Gruppe: der HTML/CSS->PDF-Weg ist ~95 % (LHT), was fehlt ist die Oberflaeche (Code-Editor mit Live-Vorschau, Site-Verwaltung, Vervollstaendigung) — das sind ED50–ED52.

MG · Gemeldete Maengel an bereits Gebautem

Nicht fehlende Faehigkeiten, sondern Dinge, die da sind und nicht so arbeiten, wie sie sollen. Sie gehen den Punkten unten vor: eine halb funktionierende Faehigkeit ist schlimmer als eine fehlende, weil der Anwender ihr vertraut.

Punkt	Was gemeldet ist	Was zu tun ist	Aufw	Stan
MG1 · Einband-Gestaltung	Der Cover-Weg (Bibliothek M1159, REST + Editor) „funktioniert nicht wie es sollte“ — die erzeugte Vorlage entspricht nicht der Erwartung	Am konkreten Fall nachmessen: Bogenmasse, Ruecken, Falzlinien, Klappen gegen ein von Hand gerechnetes Soll; die Abweichung benennen, dann beheben. Erledigt (M1319). Die Geometrie war korrekt (gegen Handrechnung nachgemessen); die Falzlinien sind eine /Separation-Sonderfarbe, und der Rasterizer las den Tint-Wert als Helligkeit — Vollton wurde reines Weiss. Die Anzeigeliste loest den Farbraum jetzt auf (IsSeparationSpace + SeparationTint)	M	
MG2 · Montageflaeche im Editor	Die Montageflaeche (M1228 Kern, M1229 Oberflaeche) „funktioniert nicht wie es sollte“	Nachstellen: Objekt auf die Flaeche ziehen, speichern, neu laden, PDF erzeugen. Erledigt (M1319). Modell und .kdoc-Rundlauf waren korrekt; die Anzeige-Felder wurden nur beim Oeffnen des Eigenschaften-Dialogs nachgezogen, nicht bei jedem Dokumentwechsel (Oeffnen/Neu/Einband/Importieren/Undo/Redo)	M	
MG3 · Editor: Seitenwechsel sehr langsam	Beim Wechseln der Seite war der Editor „extrem langsam“	Erledigt (M1317). Gemessen: 667 ms je Wechsel — der Hintergrund hielt genau EINE Seite und rechnete bei 150 dpi neu, auch beim Zurueckblattern. Jetzt ein Speicher je Seite (RenderCache, testbar) und 96 dpi	M	

MG4 · Import verhaut Fremd-PDFs ist mit M1314 erledigt: die Wortzwischenraeume wurden bedingungslos statt nach der geometrischen Luecke gesetzt, und die Vorschubbreiten der Standardschriften waren eine Pauschale.

Reihenfolge, die ich empfehle

Nicht nach Aufwand sortiert, sondern nach Nutzen je Aufwand. Die ersten vier Raenge sind zusammen kleiner als ein einziger XL-Punkt.

Rang	Was	Warum zuerst	Aufw.
1	IMG1 BMP-Decoder	Schliesst einen aktiven Fehler: .bmp wird als unterstuetzt angeboten und scheitert still	S
2	TXT1–TXT3 + TXT6 Klartext-Editor + Kodierung erkennen/konvertieren	Ersetzt Notepad ganz; der Kodierungs-Teil ist ausdrecklicher Nutzerwunsch und in keinem der Vorbilder gut geloes	S–M

Rang	Was	Warum zuerst	Aufw.
3	ED38/ED39/ED40/ED44/ED46 Oberflaeche fuer vorhandene Dienste	Schwaerzen, Signieren, Vergleichen, Aenderungsverfolgung, Kopf-/Fusszeilen gibt es schon als getestete Bibliothek — es fehlt nur der Knopf. Billigster Acrobat-/Word-Gewinn	S–M
4	IMG2 TIFF schreiben (Farbe, mehrseitig)	Der meistgeforderte Schreibweg im Druck- und Archivumfeld; heute nur 1-Bit	M
5	LPP1–LPP4 PowerPoint-Kern (Layouts, Tabellen, Formen, Aufzaehlungen)	Groesste inhaltliche Luecke der acht Vorbilder	L
6	FNT1 + FNT2 WOFF/WOFF2 schreiben + Bitmap-Emoji	Schliesst die Schreibrichtung bei Schriften und die Emoji-Darstellung	M
7	ED43 Fliesstext-Modus / ED47 Tabellenblatt-Modus	Die zwei strukturellen Editor-Luecken zu Word und Excel (Rechenkern ist da)	L
8	AG1 + AG2 Transparenz-Flattener + Preflight-Fixup	Rundet die Vorstufe ab (Pruefung ohne Korrektur ist nur die halbe Miete)	L

v4-AP · Acrobat Pro

Gruppe	Worum es geht	Aufw.	Stand
AP1 · Barrierefreiheit REPARIEREN	Auto-Tagging eines fremden, ungetaggtten PDF (Acrobat „Autotag Document“) + Reading-Order-Werkzeug. Heute nur gepueft (check_pdfua/AccessibilityAudit/veraPDF), nicht repariert.	XL	
AP2 · Prepare Form (Feld-Autoerkennung)	Formularfelder auf einem flachen PDF/Scan automatisch erkennen + anklickbar anlegen (Text/Checkbox/Radio). Heute nur erstellen/importieren/ausfuellen/XFA.	L	
AP3 · Bild in-place bearbeiten	Ein Bild in einem fremden PDF anklicken -> verschieben/skalieren/ersetzen, ohne Import-Rundlauf (das Gegenstueck zu edit_text/M1278 fuer Bilder).	L	
AP4 · Messwerkzeuge	M1325 - Annotations/RectilinearMeasurement (rechnen) + /Measure /Subtype /RL an den Formen, MCP add_measurement. Grenze: nur gleichmaessiger Massstab	M	
AP5 · Transparenz-Flattener	M1321 - Editing/PdfTransparencyFlattener, MCP flatten_transparency. Grenze: die Seite wird ganz gerastert (keine vektorielle Zerlegung)	L	
AP6 · Rich Media / 3D	Video/Audio als abspielbare RichMedia-Annotation; U3D/PRC-3D einbetten + 3D-Ansichten.	XL	(Nische)
AP7 · Editor: fremdes PDF IN-PLACE editieren	Im visuellen Editor ins gerenderte Fremd-PDF klicken und Text/Bild direkt aendern (byte-schonend), statt Import->Re-Render (ED14). Text-In-place gibt es nur als MCP-Werkzeug edit_text, nicht im Editor. Buendelt die Editor-Anbindung von AP1/AP2/AP3.	XL	

Bewusst ausserhalb (kein fehlender Code — gehosteter Dienst / Hardware, kein Offline-Aequivalent):

Punkt	Warum ausserhalb
Adobe Sign / „Zur Unterschrift senden“	Gehosteter Cloud-Dienst. Signieren/Zertifizieren/PAdES/LTV selbst ist voll da.
Scanner ansteuern (TWAIN/WIA)	Treibergebunden; Scans werden verarbeitet (OCR), aber kein Geraet getrieben.

Punkt	Warum ausserhalb
Document Cloud / Cloud-Teamarbeit	Hosting-Dienst, kein Bibliotheks-Feature.

v4-LID · InDesign

Schon da (NICHT gelistet): Masterseiten, Absatz-/Zeichen-/Objekt-/Tabellenformate, verschachtelte + GREP-Formate, Data Merge, GREP-Suche, Buch, Verpacken, verankerte Objekte, bedingter Text, OpenType, Multi-State, IDML, Ecken-Radius, Strich-/Strichlinien-Stile, Mehrspaltensatz, Textumfluss.

#	InDesign-Funktion	Stand bei dir	Aufw.
LID1	Objekt-Effekte ausser Schlagschatten: innerer Schatten, Schein (ausser/innen), abgeflachte Kante/Relief, Satin, weiche Kante (Feather)	Layout/ObjectEffects (die weiche Kante ueber eine echte PDF-Weichmaske; die uebrigen vier sind eine rechteckige Naeherung, nicht silhouettentreu - in der Klassendoku benannt; noch nicht im .kdoc gesichert)	L
LID2	Pathfinder + zusammengesetzte Pfade: Vereinen/Subtrahieren/Schnittmenge/Ausschluss; Compound Paths	(Layout/Pathfinder, ShapeKind.CompoundPath)	M
LID3	Text-Variablen <var:name> (Dateiname/Version/Kapitel/benutzerdefiniert, TextVariables.Build) M1284; laufender Kolumnentitel (FlowDocument.BuildWithRunningHeader) M1345 - letzte Seitenzahl war ueber FlowDocument.PageNumbers {{total}} bereits erledigt	M1284/M1345	M
LID4	Grundlinienraster + optischer Randausgleich (haengende Interpunktion)		M
LID5	Editor: Ausrichten & Verteilen + Duplizieren-und-versetzen (Step and Repeat)	EditorAlignDistribute/EditorStepAndRepeat + Oberflaeche. Grenze: gerechnet wird auf dem ungedrehten Rahmen, eine Gruppe zaehlt als ein Rechteck	M
LID6	Liquid Layout / alternative Layouts (adaptive Regeln fuer mehrere Seitengroessen)	M1347 (Layout/LiquidLayout.Apply): Skalieren, Neu zentrieren und Objektbasiert (feste Kanten je Achse) rechnen eine Seite auf eine andere Groesse um. Offen: die hilfslinienbasierte Regel - das Objektmodell haelt Hilfslinien nicht als benannte Groessen	L (Nische)

v4-LWD · Word (.docx)

Schon da: Lesen/Schreiben (Ueberschriften/Fett/Kursiv/Farbe/Groesse, Tabellen mit Zellfarbe/Breiten, Listen, Bilder, Hyperlinks, Schriftnamen), docxpdf, Kopf-/Fusszeile beim pdf->docx (X12), Seitenumbrueche.

#	Word-Funktion	Stand bei dir	Aufw.
LWD	Aenderungsverfolgung LESEN (w:ins/w:del, Accept-Changes-Endfassung) M1289; Markup-Ansicht + Kommentare + Schreiben noch offen	M1289	L
LWD	Fussnoten in .docx LESEN (w:footnoteReference -> Fusstext) M1286; Endnoten + Schreiben noch offen	M1286	M
LWD	Felder w:fldSimple (Cache-Wert von PAGE/DATE/REF/... anzeigen) M1288; komplexe Felder (w:fldChar) noch offen	M1288	M
LWD	Kopf-/Fusszeilen + Abschnitte beim LESEN (verschiedene Seitengroessen/Spalten je Abschnitt)	w:sectPr mit headerReference/footerReference, w:pgSz und w:cols; gestempelt ueber den vorhandenen HeaderFooterStamper	M
LWD	Formeln (OMML/Office-Math) lesen/schreiben	Docx/OmmLToLatex - gelesen und ueber den vorhandenen Formelsatz gesetzt; ohne Mathe-Schrift wird die Formel gemeldet statt geraten. Offen: OMML schreiben	L
LWD	Textrahmen/Formen/SmartArt/Diagramme	fehlt	L (Nische)

v4-LXL · Excel (.xlsx)

Schon da: Lesen/Schreiben (Zelltypen Zahl/Text/Datum/Bool, verbundene Zellen, Spaltenbreiten, Fett, mehrere Blaetter), xlsxpdf, html->xlsx, pdf->xlsx.

#	Excel-Funktion	Stand bei dir	Aufw.
LXL1	Formel-Auswertung (numerischer Rechenkern XlsxFormula: SUM/AVERAGE/MIN/MAX/COUNT/IF/ROUND/ABS/VLOOKUP + Arithmetik/Vergleiche/Bereiche) + Formeln schreiben (<f>, M1291) - Text-Funktionen/Datums-Arithmetik/benannte Bereiche/Neuberechnungs-Kette	M1291	XL
LXL2	Zellformatierung lesen + rendern: Fuellfarbe/Fett/Schriftfarbe (M1291) - Rahmen/Ausrichtung		L
LXL3	Bedingte Formatierung (cells read/write + im xlsx->pdf angewandt) + Datenpruefung/Dropdowns (list read/write, M1293) - colorScale/dataBar/iconSet + Zahl-/Datumspruefung	M1293	L
LXL4	Diagramme (Balken/Linie/Torte) im Blatt LESEN (Zeichnungs-Kette -> chart-Teil) + als natives ChartFrame in xlsx->pdf gerendert (M1296) - Diagramme SCHREIBEN + Flaeche/Streu/Ringe	M1296	L
LXL5	Benannte Bereiche (definedName) + Fensterfixierung (pane) + Autofilter (autoFilter) read/write (M1294) - Pivot	M1294	M (Nische)

#	Excel-Funktion	Stand bei dir	Aufw.
LXL6	Bilder im Blatt LESEN (Zeichnungs-Kette -> xl/media) + in xlsx->pdf gerendert (M1295) - Bilder SCHREIBEN + Formen/Diagramm-Zeichnungen	M1295	M (Nische)

v4-LFO · Apache FOP (XSL-FO)

Schon da: XF1–XF11 (Seitengeometrie, static-content Kopf/Fuss, block/inline, Tabelle, Liste, external-graphic, page-number, footnote, basic-link, XSLT, Bidi/writing-mode, column-count, Property-Vererbung).

#	FOP-Funktion	Stand bei dir	Aufw.
LFO1	fo:marker / retrieve-marker (laufende Kolumnentitel aus dem Inhalt)	M1312 (Fo/FoContext + HeaderFooterOptions.ResolvePerPage): der Kolumnentitel wird je Seite aus dem Inhalt aufgelöst, ueber den vorhandenen HeaderFooterStamper. Grenzen: retrieve-position als first/last auf der Seite, keine retrieve-boundary-Feinheiten, Markerinhalt als Klartext	M
LFO2	Zell-Spans (number-columns-spanned / number-rows-spanned) + background-color je Zelle	M1281	M
LFO3	fo:leader M1285; fo:float + echte fo:footnote M1312 (ueber die vorhandene Umflussmechanik FloatToTop/FloatToSide M234/M1022 und ParagraphWithFootnotes M991 - nichts neu gebaut). Grenzen: Float-Hoehe per Messlauf, before-Float schiebt nur Folgeinhalt, Fussnote in einer Change Bar wird gemeldet	M1312	M
LFO4	fo:bookmark-tree (explizite PDF-Lesezeichen) + Change Bars + fo:index	M1312: Lesezeichenbaum ueber FlowDocument.AddOutline/AddChild, Change Bars ueber BoxFrame.BorderLeft (kein neuer Zeichenweg), Register ueber IndexMarker/BuildWithIndexes. Grenzen: unaufloesbares Ziel wird gemeldet, Change Bar am Block statt je Zeile, Index ODER Fussnoten	M
LFO5	Weitere FOP-Ausgaben (PostScript/PCL/AFP/PNG/TIFF aus demselben FO)	M1312 (Fo/FoOutput + MCP fo_to_png/fo_to_tiff/fo_to_ps): PNG je Seite, ein mehrseitiges TIFF, PostScript Level 2. Ehrlich: das PostScript ist Raster-PostScript (Text ist darin kein Text mehr); PCL und AFP sind NICHT gebaut - es gibt keinen Baustein, auf den sie abbildbar waeren (AFP ist IBM-proprietär)	L (Nische)

v4-LTX · LaTeX (pdflatex/xelatex)

Schon da: eigene TeX-Engine W1–W28 (Makros, Abschnitte, Formatierung, 2D-Mathe mit gebuendelter Schrift inkl. align/cases/Matrizen, Tabellen, Listen, Fussnoten, TOC/Querverweise/thebibliography, geometry, babel, Silbentrennung, Knuth-Plass, \hfill/\vfill, Inline-Links).

#	LaTeX-Funktion	Stand bei dir	Aufw.
LTX1	TikZ/PGF (Zeichensprache fuer Diagramme/Grafik)	fehlt	XL (Nische, gross)
LTX2	BibTeX/biblatex (.bib parsen + Zitierstile Autor-Jahr/numerisch)	nur thebibliography + \cite[n]	L
LTX3	Theorem-Umgebungen (\newtheorem, nummeriert) M1283; weitere Mathe (\substack, kommutative Diagramme) noch offen	M1283	M
LTX4	book/report-Klasse: \part\chapter + Nummerierung (Kapitel.Abschnitt)	M1282	M
LTX5	Beamer (Praesentationsklasse)	fehlt	L (Nische)
LTX6	fontspec/XeLaTeX Systemschrift-Einbettung + microtype (Randausgleich/Font-Expansion)	Standard-14/gebuendelt	M

v4-LHT · HTML/CSS (Browser-Rendering)

Schon da: toleranter Leser, CSS-Kaskade + erweiterte Selektoren, Boxmodell, Float, position (relative/absolute/fixed), Mehrspalten (column-count), Tabellen (colspan/rowspan), Inline-SVG, Web-Fonts (opt-in Net-Add-on), JS-Subset (opt-in).

#	Browser-Funktion	Stand bei dir	Aufw.
LHT1	CSS Flexbox (display:flex, flex-grow/wrap/order/align-self)	M237/M643 (war faelschlich als Luecke gefuehrt)	L
LHT2	CSS Grid (display:grid, auto-fill)	M-Grid (war faelschlich als Luecke gefuehrt)	L
LHT3	CSS Transforms (translate/rotate/scale + transform-origin)	M656/M886; Transitions/Animations = statisches Standbild (Medium-Grenze)	M
LHT4	JavaScript - Ladezeit-Subset + skript-definierte Funktionen/Closures/Rekursion M1290; Events/Timer/fetch sind in einem statischen PDF nicht ausfuehrbar (Medium-Grenze)	M1290	XL
LHT5	@media (Typ + Breiten-Features min/max-width) M651; position:sticky als In-Fluss gerendert (Scroll ist kein PDF-Konzept)	M651 (war faelschlich als Luecke gefuehrt)	M
LHT6	CSS Custom Properties (--x/var()) + calc()/min/max/clamp	M887/M888 (war faelschlich als Luecke gefuehrt)	S

v4-LPP · PowerPoint (.pptx)

Schon da: positionierte Text-/Bildrahmen lesen+schreiben (Folienmasse, Absaetze/Laeufe mit Fett/Kursiv/Groesse/Farbe/Ausrichtung, Bilder aus ppt/media), pptx->pdf, pdf->pptx (M1205). Der Leser kennt heute genau drei Elemente (sp/pic/txBody) — alles andere unten ist im Code geprueft NICHT vorhanden.

#	PowerPoint-Funktion	Stand bei dir	Aufw.
LPP1	Layouts + Master + Platzhalter (sldLayout/sldMaster/ph): geerbte Titel-/Inhaltsrahmen, Position und Format aus dem Layout	fehlt (jede Folie steht fuer sich)	L
LPP2	Tabellen im Foliensatz (graphicFrame/tbl) lesen + schreiben	fehlt	M
LPP3	Formen-Geometrie (prstGeom: Rechteck/Ellipse/Pfeil/Stern/Callout) + Fuellung/Kontur/Verlauf/Schatten	fehlt (Formen kommen nur als Textrahmen an)	L
LPP4	Aufzaehlungen + Absatzformat (buChar/buAutoNum, Einzuege, Zeilenabstand, Tabulatoren)	fehlt (Absaetze ohne Marke/Einzug)	M
LPP5	Notizen (notesSlide) + Folien-Kommentare lesen/schreiben	fehlt	S
LPP6	Themen (theme1.xml: Farb- und Schriftschema) statt Standard-14-Abbildung	fehlt	M
LPP7	Bild-Zuschnitt (blipFill/srcRect), Gruppen (grpSp), Drehung/Spiegelung	fehlt	M
LPP8	Diagramme + SmartArt (c:chart, DrawingML-Diagramme)	fehlt	L (Nische)
LPP9	Uebergaenge/Animationen - nur fuer den pptx->pptx-Rundlauf sinnvoll (im PDF nicht darstellbar)	fehlt	M (Nische)

v4-IMG · Bildformate — Lesen und Schreiben

Schon da beim Lesen (je eigener Codec, kein Fremdpaket): JPEG, PNG, TIFF (unkomprimiert/LZW/Deflate/PackBits/CCITT G3+G4/JPEG-in-TIFF, 1–16 Bit), GIF, WebP (VP8L und VP8), HEIF/HEIC (eigener HEVC-Intra-Decoder, M1147), JPEG 2000 (EBCOT/DWT), JBIG2, CCITT.

Geschrieben werden heute nur PNG (indiziert/TrueColor), JPEG, 1-Bit-TIFF (unkomprimiert, fuer CTP) und JBIG2. Genau das ist die Luecke: die Lese-Seite ist breiter als die Schreib-Seite.

Die Matrix — welches Format in welche Richtung (im Code geprueft):

Format	Lesen	Schreiben	Bemerkung
JPEG	eigener Codec		inkl. progressiv

Format	Lesen	Schreiben	Bemerkung
PNG			indiziert + TrueColor
TIFF	voll	nur 1-Bit	Lesen: unkompr./LZW/Deflate/PackBits/CCITT G3+G4/JPEG-in-TIFF, 1-16 Bit · Schreiben: bilevel fuer CTP -> IMG2
GIF			LZW/Interlace/Transparenz -> IMG3
WebP	verlustfrei und verlustbehaftet		VP8L + VP8 (M1003) -> IMG4
HEIF/HEIC	echte Pixel (M1147)		eigener HEVC-Intra-Decoder; schreiben -> IMG7
JPEG 2000		verlustfrei (M1306) + verlustbehaftet (M1344, ohne Ratensteuerung)	eine echte Zielgroesse braucht den PCRD-Optimierer -> IMG17
WebP	verlustfrei und verlustbehaftet (M1003)	nur verlustfrei (M1301)	VP8-Encoder -> IMG16
JBIG2		x	Decoder da; Encoder fehlt -> IMG23
CCITT G3/G4			Filter-Ebene
BMP	(M1297, M1334)	(M1308, M1334)	24/32 Bit, Palettenbilder 1/4/8 Bit, BI_RLE8/BI_RLE4
TGA · Netpbm	(M1302)	(M1308)	TGA auch lauflaengen-kodiert
ICO/CUR · ANI	(M1302, M1329)	ICO/CUR (M1329)	mehrere Groessen, Klickpunkt; .ani nur lesen
PSD	Composite (M1304)	Composite (M1330)	kanalweise, PackBits/ZIP, Graustufen/RGB/CMYK
EMF/WMF	(M1305)	(M1328)	Vektor, in beide Richtungen — auch eine PDF-Seite als .emf
AVIF	Container + AV1-Sequenzkopf (M1310)	x	Bilddaten noch nicht dekodiert -> IMG6/IMG7
RAW/DNG	Sensordaten + Entwicklung (M1309)	x	inkl. verlustfreiem JPEG (SOF3); schreiben ist nicht vorgesehen
JPEG XL	x	x	-> IMG8

Kassensturz (aus dem Code gelesen, Stand M1306): von 15 lesbaren Rasterformaten (mit PSD und DNG) sind 6 schreibbar (PNG, JPEG, TIFF, GIF, WebP verlustfrei, JPEG 2000 verlustfrei). Gelesen wird also fast alles, geschrieben deutlich weniger — die folgenden Punkte schliessen genau diese Luecke.

#	Bildformat-Funktion	Stand bei dir	Aufw.
IMG1	BMP lesen - Fehler: .bmp steht in DocumentConverter.SupportedExtensions, aber IRasterImage.Parse kennt keine BMP-Signatur -> die Datei wird als unterstuetzt angeboten und schlaegt dann still fehl	Bug, Decoder fehlt ganz	S
IMG2	TIFF schreiben in Farbe + mehrseitig (8/16 Bit, Graustufen/RGB/CMYK, LZW/Deflate/PackBits)	nur 1-Bit unkomprimiert (TiffBilevel)	M
IMG3	GIF schreiben (Palette/Transparenz, optional animiert)	nur lesen	M
IMG4	WebP schreiben (VP8L verlustfrei; VP8 verlustbehaftet)	nur lesen	L
IMG5	JPEG 2000 schreiben (EBCOT-Encoder; der MQ-Coder ist bereits da)	M1306 (Jpeg2000Encoder + convert_image): verlustfrei ueber die reversible 5/3-Wavelet, ein Tile/eine Schicht/LRCP. Von openjpeg als pixelgenau bestaetigt. Grenze: der verlustbehaftete 9/7-Weg braucht eine Ratensteuerung und fehlt	L
IMG6	AVIF lesen (AV1-Intra-Decoder) — in Scheiben wie der HEVC-Decoder (M1111–M1147)	M1310 (Av1BitReader/Av1Obu/Av1SequenceHeader/Av1Configuration): OBU-Schicht, Sequenzkopf (Masse, Bittiefe, 4:2:0/4:2:2/4:4:4/monochrom, Werkzeugschalter) und die av1C-Box, von libavif gegengelesen; HeifDecoder meldet jetzt die echte kodierte Groesse und Farbanordnung. Offen: Rahmenkopf, Symbol-Decoder (CDF), Partitionsbaum, die Intra-Modi, Transformationen, CDEF/Wiederherstellung/Filmkorn - also die Pixel	XL
IMG7	HEIF/AVIF schreiben (HEVC- bzw. AV1-Intra-Encoder)	fehlt	XL (Nische)
IMG8	JPEG XL lesen + schreiben	fehlt	XL (Nische)
IMG9	EMF/WMF lesen -> PDF-Vektorinhalt. Praxisrelevant: Office-Dokumente tragen Grafiken oft als Metafile	M1305 (Metafile-Modul + metafile_to_pdf): Rechteck/Ellipse/Streckenzug/Vieleck/Linie/Text mit Farben und Schriftgroesse, beide Formate. Grenzen: Bitmap-Aufzeichnungen, Pfad-/Clip-Bereiche, Welt-Transformationen - gemeldet, nie still	L
IMG1	PSD lesen (zusammengesetztes Bild, ohne Ebenen-Editierung)	fehlt	M (Nische)
IMG1	ICO / TGA / PNM (PBM,PGM,PPM) lesen	fehlt	S (Nische)

#	Bildformat-Funktion	Stand bei dir	Aufw.
IMG1	RAW/DNG lesen (Bayer-Demosaicing)	M1309 (DngReader + LosslessJpegDecoder + BayerDemosaic): Sensordaten aus dem Unterverzeichnis (nicht die Vorschau!), unkomprimiert und verlustfrei JPEG (SOF3), Dunkel-/Saettigungswert, Aufnahme-Weissabgleich, Kameramatrix, bilineares Entmosaiken. Von LibRaw gegengelesen. Grenzen: verlustbehaftetes DNG und die herstellereigenen Rohformate (CR2/NEF/ARW) fehlen; Kalibrierungs-/Vorwaertsmatrix und Kennlinien-Tabelle werden nicht ausgewertet	L (Nische)
IMG1	Werkzeug convert_image (Format->Format ueber die eigenen Codecs, mit Kompressions-/Qualitaetswahl) + Pruefstand-Seite	fehlt (setzt IMG2-IMG5 voraus)	S
IMG1	EXIF lesen und schreiben (JPEG-APP1, TIFF-Tags, PNG-eXIf): Aufnahmezeit, Kamera/Objektiv, Belichtung, ISO, Ausrichtung, GPS — auch beim Wandeln mitnehmen statt verlieren	fehlt ganz (der JPEG-Leser kennt nur APP14; PdfSanitizer kann Metadaten nur entfernen)	M
IMG1	IPTC/XMP im Bild lesen und schreiben (JPEG-APP13/Photoshop-Ressourcen, XMP-Paket): Titel, Beschreibung, Urheber, Rechte, Stichwoerter — die Grundlage fuer Katalog und Sidecar	fehlt (XMP gibt es nur fuer PDF-Dokumente, M71)	M
IMG1	WebP verlustbehaftet schreiben (VP8-Encoder): Intra-Praediktion, DCT, Quantisierung, boolescher Entropie-Codierer, Schleifenfilter. Der Decoder ist seit M1003 da und ist zugleich das Gegenstueck zum Pruefen	nur verlustfrei (M1301)	XL
IMG1	JPEG 2000 verlustbehaftet schreiben: 9/7-Wavelet + Quantisierung + Ratensteuerung (welche Codierdurchlaeufer fallen weg — der PCRD-Optimierer). Alles Uebrige steht seit M1306	M1344: 9/7-Wavelet + Quantisierung je Teilband (Exponent/Mantisse im QCD), Stufe 1..100, auch in convert_image (lossy); openjpeg liest es mit hoechstens 3 Abweichung. Offen: die Ratensteuerung - die Stufe waehlt die Schrittweite, nicht eine Zielgroesse	L
IMG1	BMP, TGA und Netpbm schreiben	M1308 (BmpEncoder/TgaEncoder/PnmEncoder + convert_image): BMP 24/32 Bit bottom-up mit Zeilenauffuellung, TGA unkomprimiert und laeuflaengen-kodiert, Netpbm P4/P5/P6. Von Pillow als pixelgenau bestaetigt. Grenzen: keine Palettenbilder, keine BMP-RLE, Netpbm nur binaer	S
IMG1	ICO/CUR schreiben	M1329 (IcoEncoder + ImageFormat.Ico): mehrere Groessen in EINER Datei, PNG- und DIB-Eintraege mit AND-Maske, von Pillow pixelgenau gegengelesen. Grenzen: keine Palettenbilder, Kante hoechstens 256	M
IMG2	PSD schreiben (Composite)	M1330 (PsdEncoder + ImageFormat.Psd): kanalweise, unkomprimiert/PackBits/ZIP, Graustufen/RGB(+Alpha)/CMYK invertiert. Von Pillow pixelgenau gegengelesen (ausser ZIP, das Pillow im Composite-Abschnitt nicht kann). Dabei ein stiller Leser-Fehler behoben: der ZIP-Zweig las jeden Kanal ab derselben Stelle. Grenzen: keine Ebenen, 8 Bit, kein PSB, kein ICC	M
IMG2	EMF/WMF schreiben — die Vektor-Gegenrichtung zu M1305	M1328 (MetafileWriter + PdfToMetafile + pdf_to_metafile): beide Formate aus dem Zeichenmodell, und eine PDF-Seite ueber die Anzeigeliste (echte Pfade mit aufgeloeseten Farben, nicht die blossen Rechtecke des Grafik-Extraktors). Dabei zwei stille Leser-Fehler behoben: Pinsel und Stift faerbten einander, und die Strichstaerke ging verloren. Grenzen: keine Bilder, kein Beschnitt/Verlauf/Transparenz, Bezierkurven werden in Strecken zerlegt	M

#	Bildformat-Funktion	Stand bei dir	Aufw.
IMG2	.j2k im Erkennungsweg - Fehler derselben Klasse wie IMG1: RasterImage.Parse kannte JPEG 2000 nicht, obwohl Decoder und Encoder da waren	M1307: Codestream (FF4F FF51) und JP2-Box erkannt; convert_image/images_to_pdf/Dokument-Import nehmen .j2k jetzt als Quelle	S
IMG2	JBIG2 schreiben (generische Region, arithmetisch; optional Symbol-Woerterbuch fuer Text). Praxisnutzen: gescannte Schwarz-Weiss-Seiten im PDF deutlich kleiner als CCITT G4. Der Decoder ist da	M1345 (Jbig2Encoder): der eingebettete Strom - Seiteninformation, unmittelbare generische Region, Seitenende; kleiner als der eigene CCITT-G4-Schreiber. + M1346 (Einbettung ins PDF ueber PdfBilevelRecompressor + recompress_bilevel; dabei zwei Fehler gefunden - das /Decode gehoert NICHT dazu, und der eigene Leseweg uebergang es). Offen: das Symbol-Woerterbuch (der Trumpf bei Textseiten)	L
IMG2	Mauszeiger vollstaendig (.cur/.ani)	M1329 (IcoDecoder.DecodeAll/IsCursor + AniDecoder): Klickpunkt lesen und schreiben, .ani mit allen Einzelbildern und ihren Standzeiten aus dem rate-Block; .ani auch im Erkennungsweg. Grenze: .ani wird nicht geschrieben, rohe DIB-Blöcke (bfAttributes Bit 1) werden uebersprungen statt geraten	S-M
IMG2	macOS-Symbole (.icns) lesen und schreiben: die Typ-Blöcke (ic04...ic14), je nach Groesse roh, PackBits-komprimiert (is32+s8mk) oder als eingebettetes PNG/JP2	x komplett	M
IMG2	Symbole aus Programmen holen (.exe/.dll): den PE-Ressourcenbaum lesen und die RT_GROUP_ICON/RT_ICON-Eintraege zu einer .ico zusammensetzen — der uebliche Weg, an das Symbol einer Anwendung zu kommen	x	M

Detailtiefe je Format. Ein Format „koennen" heisst nicht, jede seiner Spielarten zu koennen. Der folgende Durchgang ist am Code gemessen und nennt, was je Format noch fehlt — meist Kompressions- oder Struktur-Varianten, die in echten Dateien vorkommen.

#	Was fehlt	Warum es vorkommt	Aufw
IMG2	TIFF: LZW und CCITT G4 schreiben; BigTIFF lesen. Berichtigung: gekachelte TIFFs (TileWidth/TileLength) und die planare Ablage (PlanarConfiguration 2) werden seit M122 bereits gelesen — das stand hier faelschlich als offen	M1331 (LzwEncoder + CcittFaxEncoder + TiffCompression.Lzw + TiffBilevelCompression.Group4): beide Codierer von Pillow pixelgleich gegengelesen; der LZW-Codierer dient zugleich PDFs /LZWDecode. Grenzen: kein Praediktator, nur G4 (nicht G3). M1332: BigTIFF (Version 43) wird gelesen - 8-Byte-Versaetze, 20-Byte-Eintraege, die Typen LONG8/SLONG8/IFD8, auch im Erkennungsweg; Fixture aus einem fremden Schreiber (Pillow). Grenze: die API nimmt ReadOnlyMemory<byte>, also passt eine Datei ueber 2 GB gar nicht erst hinein - BigTIFF heisst hier „BigTIFF-Struktur", nicht „Datei ueber 4 GB"	M-L
IMG2	PNG: Adam7-Interlace und 16 Bit je Kanal schreiben; dazu APNG lesen und schreiben	M1333 (PngEncoder.WriteTrueColor16 + interlace-Schalter): 16 Bit in RGB und Graustufen mit und ohne Deckkraft, Adam7 in allen sieben Durchgaengen auch bei ungeraden Massen. Von Pillow bestaetigt - das 16-Bit-Graustufenbild kommt als I;16 mit exakt den geschriebenen Werten zurueck, das verschachtelte pixelgleich zum glatten. M1336: APNG lesen und schreiben (siehe IMG30)	M

#	Was fehlt	Warum es vorkommt	Aufw
IMG2	JPEG: progressiv schreiben; 12 Bit und arithmetische Kodierung lesen	M1338 (12 Bit): SOF1 mit Praezision 12 wird gelesen - Pegelverschiebung 2048, Klemmung auf 4095, danach auf 8 Bit heruntergerechnet (benannte Grenze, DecodedImage haelt Bytes). Fixture von libjpeg-turbo. + M1343 (progressiv schreiben: SOF2, Gleichanteile MCU-verschraenkt, dann je Komponente die Wechselanteile; Spektralauswahl, keine sukzessive Annaeherung; libjpeg liest es als progressiv und pixelgleich). Offen: die arithmetische Kodierung lesen	M-L
IMG3	GIF und WebP: Animationen — mehrere Einzelbilder lesen und schreiben	M1335 (GIF, GifDecoder.DecodeAnimation + GifEncoder.EncodeAnimation): alle Einzelbilder als fertig zusammengesetzte Vollbilder (Ausschnitt, Entsorgungsart 0/1/2/3, Transparenz), Standzeiten und Wiederholungszahl in beide Richtungen; von Pillow in beide Richtungen bestaetigt. Grenze: der Schreiber optimiert nicht auf Ausschnitte. M1336: APNG (Content/ApngImage) in beide Richtungen - acTL/fcTL/fdAT, Ausschnitt an seiner Stelle, Entsorgungs- und Mischart, Standzeit als Bruch, echte Deckkraft; ein geschriebenes APNG bleibt ein gueltiges PNG, das das erste Einzelbild zeigt. Von Pillow in beide Richtungen bestaetigt. M1337: WebP-Animation (WebPDecoder.DecodeAnimation + WebPEncoder.EncodeAnimation) - VP8X/ANIM/ANMF mit eingeschachteltem Bildblock, Stelle in Zweierschritten, Masse um eins vermindert, Misch- und Entsorgungsbit; von Pillow in beide Richtungen bestaetigt. Grenze: geschrieben wird nur verlustfrei (der VP8-Codierer ist IMG16)	M
IMG3	BMP: Lauflaengen-Kompression und Palettenbilder	M1334 (BmpEncodeOptions + RLE im BmpDecoder): BI_RLE8/BI_RLE4 in beide Richtungen, Palettenbilder mit 1/4/8 Bit. Von Pillow pixelgleich gegengelesen. Zwei Befunde: mein spezifikationskonformes RLE4 mit ungerader Absolutlauf-Laenge wird von verbreiteten Lesern falsch verstanden (sie ruecken um Bytes statt Bildpunkte vor) - deshalb schreibe ich nur gerade Laengen; und die Farben werden nicht quantisiert, zu viele Farben werden benannt abgelehnt	S
IMG3	JPEG 2000: Kacheln, mehrere Qualitaetsschichten, kleine Precincts lesen und schreiben; dazu die JP2-Box-Datei schreiben	M1341 (JP2-Behaelter: Signaturbox/ftyp/jp2h/jp2c; .jp2 und .j2k getrennt) + M1342 (Kacheln lesen und schreiben - je Kachel eigene Wavelet und eigene Teilbaender, Randkacheln kleiner; openjpeg liest das Ergebnis pixelgenau). Offen: mehrere Teilstuecke je Kachel, mehrere Qualitaetsschichten, kleine Precincts; im Behaelter res /ICC/cdef	L
IMG3	HEIF: mehrteilige Bilder (grid), Deckkraft-Ebene und Tiefenkarten lesen	M1339 (grid: Kachelgitter zusammengesetzt - Zeilen/Spalten minus eins, Kachelreihenfolge aus iref/dimg) + M1340 (Deckkraft-Ebene und Tiefenkarte ueber iref/auxl + die auxC-URN; die Deckkraft wird angewandt, die Tiefenkarte als DepthMap geliefert). Benannte Grenze: bei einem Kachelgitter wird die Deckkraft nicht angewandt	L

v4-FNT · Schriften — alle Arten, beide Richtungen

Schon da: TrueType(glyf), OpenType/CFF, bare CFF, CID-keyed CFF, CFF2, Type1(PFB/eexec), Type3, Type42, WOFF, WOFF2, Variable Fonts (fvar/gvar/avar/HVAR inkl. Instanzieren), Farbfonts COLR/CPAL v0+v1, MATH-Tabelle, GSUB/GPOS-Shaping (Arabisch/Devanagari/Brahmi/Thai/Khmer/Myanmar/CJK). Einbetten+Subsetten in allen genannten Formaten; Export aus PDF als .ttf/.otf/.pfb/.pfa (M990), Reparieren/Ersetzen (M952/M953).

Die Matrix — welche Schriftart in welche Richtung (im Code geprüfert):

Art	Lesen/Rendern	Einbetten/Subsetten	Als Datei ausgeben
TrueType (glyf)			.ttf byte-treu
OpenType/CFF			.otf
bare CFF (Type1C/CIDFontType0C)			-> OTF gewrappt
CID-keyed CFF		subsetting (M291)	
CFF2 (variabel)		instanziert -> statisch	
Type1 (PFB/eexec)			.pfb/.pfa
Type3 (Prozedur-Glyphen)		erzeugbar	— (kein Dateiformat)
Type42 (PostScript)			— (kein Dateiformat)
Variable Fonts (fvar/gvar/avar/HVAR)		instanziert	
Farbfonts COLR/CPAL v0+v1			
WOFF / WOFF2		(M1311)	Container-Umpacken in jeder Richtung
Farbfonts CBDT/EBDT/sbix (Bitmap-Emoji)			-> FNT2
Farbfonts SVG-in-OpenType			-> FNT3
TrueType-Hinting (fpgm/prep-Bytecode)		—	- -> FNT4
Multiple Master · BDF/PCF/.fon/.fnt			-> FNT6/FNT7

#	Schrift-Funktion	Stand bei dir	Aufw.
FNT1	WOFF + WOFF2 schreiben (Encoder; Brotli/zlib sind in der BCL)	M1311 (WoffEncoder/Woff2Encoder): sfnt -> WOFF (zlib je Tabelle) und -> WOFF2 (ein Brotli-Strom), mit korrekter urspruenglicher Groesse; Rundlauf ueber die eigenen Entpacker belegt. Grenze: kein WOFF2-Transform der glyf/loca-Tabellen (die Datei ist dadurch groesser als eine von fontTools, aber gueltig)	M

#	Schrift-Funktion	Stand bei dir	Aufw.
FNT2	Bitmap-Farbfonds CBDT/CBLC + sbix lesen und zeichnen	M1323 (Fonts/ColorBitmapFont, PageCanvas.DrawBitmapColorText) - sbix nur PNG	M
FNT3	SVG-in-OpenType (SVG -Tabelle) rendern	M1326 (Fonts/ColorFontSvg, PageCanvas.DrawSvgColorText)	M
FNT4	TrueType-Hinting (Bytecode-Interpreter fpgm/prep/glyf-Instruktionen) fuer scharfe kleine Groessen im Rasterizer	Fonts/TrueTypeHinting: Stack, Achsenwahl, MDAP/MIAP/SHP/IUP, FDEF/CALL, NPUSHB/PUSHW und die cvt-Tabelle. Eine unbekannte Instruktion bricht benannt ab, statt zu raten. Offen: der volle Befehlssatz und die Anbindung an den Rasterizer	L
FNT5	Werkzeug convert_font - Schriftdatei in alle Richtungen wandeln (tftotfwoffwoff2pfb/pfa) + Pruefstand-Seite	M1311 (Fonts/FontConverter + MCP convert_font): Container umpacken in jeder Richtung, bare-CFF in einen OpenType-Rahmen, pfb pfa. Grenze: es werden KEINE Umrisse umgerechnet - glyf CFF und Type1 sfnt werden benannt abgelehnt statt kaputt geschrieben; kein Subsetting. Pruefstand-Seite fehlt noch	S
FNT6	Type1 Multiple Master (Instanzieren wie bei Variable Fonts)	Fonts/Type1MultipleMaster: aus einer Achsposition wird ein fester Schnitt gerechnet	M (Nische)
FNT7	Alt-Bitmapschriften BDF/PCF/.fon/.fnt lesen	fehlt	S (Nische)

v4-FED · Schriften — jede Art lesen, schreiben und ineinander wandeln, plus Editor

Der Wunsch: moeglichst alle Schriftarten, die es gibt und die benutzt werden — Web, Druck und Betriebssysteme — lesen, schreiben und in jede andere Art wandeln, ohne Fremdprodukt (kein FontForge, kein fonttools). Dazu ein Editor, um bestehende Schriften zu aendern und neue zu bauen.

Was schon da ist (nicht doppelt bauen): Leser fuer TrueType(glyf), OpenType/CFF, bare CFF, CID-keyed CFF, CFF2, Type1(PFB/eexec), Type3, Type42, WOFF, WOFF2, Variable Fonts (inkl. Instanzieren) und Farbfonds COLR/CPAL v0+v1; Umriss-Auslesung (GlyphBoundsReader), sfnt-Schreiber (SfntWriter, SfntTableBuilders), CFF->OpenType-Verpackung, Subsetter fuer TrueType und CFF, Type1FontConverter (-> PFB/PFA), Cff2StaticFontBuilder und FontExporter (M990), der eine eingebettete Schrift bereits als .ttf/.otf/.pfb herausschreibt und fehlende Tabellen ergaenzt.

Die Format-Matrix — wo steht was

Format	Wofuer	Lesen	Schreiben
TTF (sfnt/glyf)	Web, Druck, System		(SfntWriter)

Format	Wofuer	Lesen	Schreiben
OTF (sfnt/CFF)	Web, Druck		(CffOpenTypeWrapper)
WOFF	Web		-> FNT1
WOFF2	Web		-> FNT1
Variable Fonts (fvar/gvar/HVAR)	Web, Druck	inkl. Instanzieren	nur als statische Instanz -> FED9
CFF2	Druck		nur statisch -> FED9
CID-keyed CFF (ROS)	CJK-Druck		-> FED10
Type1 (PFB/PFA)	Alter Druck		(Type1FontConverter)
Type3	PDF-intern		(PDF-Erzeugung)
Type42	PostScript		-> FED11
TTC/OTC (Sammlungen)	System	erkannt (SystemFonts)	-> FED12
COLR/CPAL (Farbe)	Web, System	v0+v1	-> FED13
CBDT/EBDT · sbix (Bitmap-Emoji)	System	-> FNT2	-> FNT2
SVG-in-OpenType	Web	-> FNT3	-> FED13
EOT	Web (alt, IE)	-> FED14	-> FED14
SVG-Fonts	Web (abgeloeset)	-> FED14	-> FED14
BDF · PCF · PSF	Linux/Konsole	-> FNT7	-> FED15
FON/FNT	Windows (alt)	-> FNT7	-> FED15
dfont / Suitcase	macOS	-> FED16	-> FED16
UFO · designspace	Quellformat der Editoren	-> FED17	-> FED17
SFD (FontForge) · TTX (fonttools-XML)	Austausch	-> FED18	-> FED18
AFM/PFM (Metriken)	Alter Druck	-> FED19	-> FED19

Die Punkte

#	Funktion	Aufw.
FED1	Glyphen-Umriss ändern — Punkte/Kurven lesen, bearbeiten und zurückschreiben (glyf und CFF-Charstrings). Heute sind Umriss nur lesbar; das ist die Grundlage fuer fast alles Uebrig	L

#	Funktion	Aufw.
FED2	Neue Schrift von Grund auf: Glyphen aus Umrissen (auch aus vorhandenem SVG-/Vektor-Inhalt), Metriken, cmap, Namen, OS/2 — als .ttf/.otf schreiben	L
FED3	Wandeln in jede Richtung - die Matrix oben schliessen: TrueType CFF (quadratisch kubisch umrechnen), Type1 -> OpenType, Type42 -> sfnt, CFF2 -> CFF, Variable -> Instanz, sfnt -> WOFF/WOFF2	L
FED4	Metriken und Kerning bearbeiten: Vorschubbreiten, Seitenraender, kern/GPOS-Paare anlegen und aendern	M
FED5	Zeichenzuordnung bearbeiten: cmap anlegen/aendern (Format 4/6/12), Glyphennamen (post) setzen	M
FED6	Zusammenfuehren/Aufteilen: Glyphen aus mehreren Schriften in eine, oder eine Teilmenge herausloesen — mit Umbenennen und Neu-Zuordnen (mehr als Subsetting)	M
FED7	Schrift-Editor im Pruefstand: Glyphe waehlen, Umriss sehen und ziehen, Metriken setzen, Vorschau, Speichern — die Oberflaeche ueber FED1–FED6	L
FED8	Pruefer fuer selbst gebaute Schriften: Pflichttabellen, Pruefsummen, Metrik-Widersprueche — das Gegenstueck zu FontHealthChecker	M
FED9	Variable Fonts SCHREIBEN: fvar/gvar/avar/HVAR erzeugen, also aus mehreren Schnitten eine variable Schrift bauen (nicht nur eine Instanz herausrechnen)	XL
FED10	CID-keyed CFF schreiben (ROS, FDArray/FDSelect) — die Form, in der CJK-Schriften ausgeliefert werden	L
FED11	Type42 schreiben (sfnt in PostScript verpackt) — fuer die PostScript-Ausgabe	S
FED12	Sammlungen (TTC/OTC) lesen und schreiben: mehrere Schriften mit geteilten Tabellen in einer Datei	M
FED13	Farbfonts schreiben: COLR/CPAL v0+v1 und SVG -Tabelle erzeugen (Lesen ist da bzw. FNT3)	L
FED14	Web-Altlasten: EOT und SVG-Fonts lesen und schreiben — beide abgeloeset, aber in Bestaenden vorhanden	M
FED15	Bitmapschriften schreiben: BDF/PCF/PSF und FON/FNT (Lesen ist FNT7)	M
FED16	macOS-Bestaende: dfont/Suitcase (Resource-Fork) lesen und schreiben	M
FED17	UFO + designspace lesen und schreiben — das Quellformat, in dem Schriften entworfen werden; damit laesst sich mit anderen Editoren zusammenarbeiten	M
FED18	Austauschformate: FontForge-SFD und fonttools-TTX (XML) lesen und schreiben	M
FED19	Metrikdateien AFM/PFM lesen und schreiben (gehoeeren zu Type1)	S
FED20	MCP + Pruefstand: convert_font (deckt sich mit FNT5), font_info, font_edit und ein Reiter fuer den Editor	S

Ehrlich zur Verifikation: der ueberzeugende Beleg fuer eine selbst gebaute Schrift ist, dass ein fremder Renderer sie liest (HarfBuzz/FreeType, wie schon bei den OpenType-Attributen M1087). Nach der Projektregel geschieht das einmalig beim Bauen; im Repo bleibt die eingebackene Erwartung.

Ehrlich zur Vollstaendigkeit: „alle Schriftarten“ heisst hier alle, die im Web, im Druck und in Betriebssystemen tatsaechlich vorkommen — die Matrix oben zaehlt sie auf. Firmeneigene

Editor-Formate (FontLab VFB/VFC, Glyphs .glyphs) sind nicht dokumentiert; sie liessen sich nur durch Nachbau aus Beispieldateien erschliessen und stehen bewusst nicht in der Liste.

v4-TXT · Klartext / Notepad

Schon da: Dateien lesen/schreiben ueber MCP (read_file/write_file, M1248/M1255), Text->PDF, RTF/Markdown/HTML.

Was fehlt, ist der Editor selbst: keine der Pruefstand-Seiten ist ein Klartext-Editor.

#	Notepad-Funktion	Stand bei dir	Aufw
TXT1	Klartext-Editor als Pruefstand-Seite: oeffnen/bearbeiten/speichern, Zeilennummern, Suchen+Ersetzen, Gehe-zu-Zeile	M1320 (PlainText/TextLines + TextSearch in der Bibliothek, Seite /texteditor)	M
TXT2	Kodierung ERKENNEN — BOM + Heuristik mit Zuverlaessigkeitsangabe	M1313 (neues Blatt-Modul PlainText: SingleByteCharsets traegt die Zeichentabellen als Daten [Zero-NuGet - GetEncoding(1252)/CP850/MacRoman sind NICHT in-box], TextEncodingDetector erkennt ueber BOM oder Heuristik) + MCP detect_encoding. Ehrlich benannte Grenzen: ISO-8859-1 und Windows-1252 sind im Bereich 0xA0-0xFF Zeichen fuer Zeichen gleich und aus den Bytes NICHT unterscheidbar (die Erkennung meldet Windows-1252 mit hoechstens 0,75); kurzer Text traegt kaum Aussage; auf westeuropaeische Sprachen zugeschnitten	S
TXT3	Zeilenenden CRLF/LF/CR erkennen + umstellen (+ MCP-Werkzeug)	M1315 (PlainText/LineEndings + convert_line_endings) - gezaehlt wird je Zeilenumbruch, ein CRLF ist EIN Zeilenende	S
TXT4	Grosse Dateien (virtualisiertes Laden/Anzeigen ueber 100 MB, ohne alles im Speicher)	PlainText/LargeTextFile: Zeilen-Index ueber Byte-Versaetze, Zeilen einzeln nachladen, abschnittweise suchen mit Ueberlappung an den Grenzen. Grenze: ein Muster ueber mehr Zeilen als die Ueberlappung wird nicht gefunden	M
TXT5	Syntax-Hervorhebung + Klammerpaare	M1324 (JSON/XML/CSS/Markdown) + C# (gewoehnliche, woertliche, interpolierte und Roh-Zeichenketten, Doku-Kommentare)	M
Kodierung KONVERTIEREN — eine Datei in eine gewaehlte Zielkodierung schreiben (BOM ja/nein, Zeilenenden mitziehen), einzeln oder als Stapel ueber einen Ordner mit Bericht je Datei	M1315 (TextEncodingConverter/TextEncodingBatch + convert_encoding)	M	

v4-AG · Apogee-Rest (Druckvorstufe)

Schon da (v3-AP, abgeschlossen): Halbton-Screening->1-Bit-Platte, Trapping, GCR/UCR/TAC, JDF/JMF-Ticketing, Contract-Proof-Datei, Ausschliessen, Farbmanagement, Preflight-Pruefung.

#	Vorstufen-Funktion	Stand bei dir	Aufw
AG1	Transparenz-Flattener (Live-Transparenz verflachen fuer alte RIPs) — identisch mit AP5	M1321 (Editing/PdfTransparencyFlattener, MCP flatten_transparency)	L
AG2	Preflight-Fixup (korrigieren statt nur melden)	M1322 (Editing/PreflightFixup, MCP preflight_fix) - befundgetrieben: repariert nur, was der Preflight meldet	L
AG3	PDF/VT (variabler Datendruck) — die /DPart-Hierarchie ist seit M1167 da, es fehlen Metadaten + Regelwerk	PdfVtPartMetadata/PdfVtVersion beim Erzeugen, PdfVtCompliance beim Pruefen (setzt PDF/X-4 voraus). Grenze: ob die Blaetter inhaltlich je einen Datensatz treffen, bleibt ungeprueft	L
AG4	Normalizer (fremdes PDF in EINEM Lauf auf einen definierten Ausgabezustand bringen)	M1318 (Editing/PdfNormalizer, MCP normalize_pdf)	M

v4-ED · Editor — Paritaet zu den Vorbildern

Der Editor selbst (nicht die Bibliothek): was die Oberflaeche gegenueber InDesign, Acrobat Pro, Word, Excel, PowerPoint, Dreamweaver und Notepad noch WENIGER kann. Im Code der Editor-Seite geprueft (Editor.razor 7170 Zeilen + EditorService.cs 4681 Zeilen), nicht aus dem Gedaechnis.

Schon da (NICHT gelistet): WYSIWYG-Canvas mit Live-Darstellung, Inline-Textbearbeitung, Undo/Redo, Menue-/Kontroll-/Statusleiste, Rechtsklick-Menue, Musterseiten + Doppelseiten + Miniaturen, Lineal-Hilfslinien, Montageflaeche, Ebenen/OCG, Farbfelder mit eigenem Farbwaehler, Verlaeufe, Absatzformate, Verknuepfungs-Palette, Formularfeld-Bedienung, Lesezeichen, Tabellen mit Zelltext/-farben, Drehung, Suchen+Ersetzen, Rechtschreibpruefung auf Knopfdruck, Seiten-Organisierer mit Drag&Drop, Dokument-Import/-Export, PDF-Export-Dialog.

#	Vorbild	Editor-Funktion	Stand bei dir	Aufw
ED30	InDesign	Zeichenformat-Palette (eigene Absatz- UND Zeichenformate anlegen/verwalten/bearbeiten)	nur 4 feste Absatzformate (M1037), Zeichenformate im Editor gar nicht	M
ED31	InDesign	Ausrichten & Verteilen + Duplizieren-und-versetzen — deckt LID5	fehlt (kein Align/Distribute im Code, nur Smart Guides)	M
ED32	InDesign	Objekt-Effekte-Palette (weiche Kante, Schein, Relief, Satin) — deckt LID1	fehlt (nur Schlagschatten + Weichzeichnen)	M
ED33	InDesign	Story-Editor (Fliesstext-Fenster getrennt vom Layout, fuer langen Text)	fehlt	M

#	Vorbild	Editor-Funktion	Stand bei dir	Aufw
ED34	InDesign	Verknuepfte statt eingebettete Bilder + Verknuepfungen aktualisieren/ersetzen/fehlende finden	Bilder werden immer eingebettet	L
ED35	InDesign	Live-Preflight-Panel im Editor (die Pruefung gibt es als Dienst, nicht als Palette)	fehlt	M
ED36	InDesign	Softproof-/Ueberdrucken-Vorschau im Editor (ICC-Engine + Separationsvorschau sind da)	fehlt	M
ED37	Acrobat	Kommentar-/Markup-Werkzeuge im Editor (Notiz, Hervorheben, Formen, Freitext) — das Annotations-Modul ist voll da	fehlt (kein Annotations-Bezug im Editor-Code)	M
ED38	Acrobat	Schwaerzen im Editor (RedactionEditor gibt es als Dienst, ED17)	keine Oberflaeche	M
ED39	Acrobat	Signieren im Editor (SignatureEditor gibt es als Dienst, ED18)	keine Oberflaeche	M
ED40	Acrobat	Zwei PDFs vergleichen im Editor (PdfComparer ist da)	keine Oberflaeche	S
ED41	Acrobat	Tag-Baum + Leseordnung bearbeiten (Barrierefreiheit) — die Editor-Seite von AP1	fehlt	L
ED42	Acrobat	Fremdes PDF in-place editieren (ins gerenderte PDF klicken statt Import->Re-Render) = AP7	nur als MCP-Werkzeug edit_text (M1278)	XL
ED43	Word	Fliesstext-Modus: ein durchgehendes Textdokument, bei dem beim Tippen neue Seiten entstehen (heute rahmenbasiert wie InDesign)	fehlt - der groesste Unterschied zu Word	L
ED44	Word	Aenderungsverfolgung + Kommentare im Editor (TrackedText/EditorialNote sind da)	keine Oberflaeche	M
ED45	Word	Rechtschreibung beim Tippen (rote Wellenlinie) + Wort-/Zeichenzahl in der Statusleiste	nur Pruefung auf Knopfdruck; Zaehlung nur als Bibliotheksdienst	M
ED46	Word	Kopf-/Fusszeilen im Editor bearbeiten (der Stempel-Dienst HeaderFooterStamper ist da)	keine Oberflaeche	M
ED47	Excel	Tabellenblatt-Modus: Zellgitter, Formeleingabe, mehrere Blaetter — der Rechenkern XlsxFomula (M1292) und Zellformate/Diagramme (M1291–M1296) sind da	fehlt (nur TableFrame als Layout-Objekt)	L
ED48	PowerPoint	Folien-Modus (Foliensortierung, Layouts, Platzhalter)	fehlt	L
ED49	PowerPoint	Praesentieren (Vollbild-Vorfuehrung, Referentenansicht, Notizen)	fehlt	M
ED50	Dreamweav	HTML/CSS-Code-Editor mit Live-Vorschau (Split-View CodeAnsicht) - der HTML->PDF-Weg ist voll da	fehlt	L
ED51	Dreamweav	Projekt-/Site-Verwaltung (Dateibaum, Ressourcen, Verweise pruefen) — die Dateiwerkzeuge (M1248/M1255) sind da	fehlt	M
ED52	Dreamweav	Code-Vervollstaendigung + Validierung (HTML/CSS)	fehlt	M
ED53	Notepad	Klartext-Tab im Editor (siehe TXT1)	fehlt	M

v4-ARC · Archive packen und entpacken — ohne Fremdkomponente (Nutzerwunsch)

Ziel: ZIP, RAR, 7z und die auf Mac und Linux ueblichen Formate im Kern lesen und schreiben,

ohne ein Fremdprodukt. Praxisnutzen im Projekt selbst: .docx/.xlsx/.pptx/.epub/.idml/.xml

und .kdoc sind ZIP-Behaelter, und ein Dokumenten-Werkzeug muss Sammelarchive oeffnen koennen.

Was die .NET-Basis schon mitbringt (kein NuGet noetig, zaehlt also nicht als Fremdkomponente):

ZipArchive (Deflate), GZipStream, DeflateStream, ZLibStream, BrotliStream und seit .NET 7

System.Formats.Tar. Alles Weitere unten fehlt und muss selbst gebaut werden.

#	Format / Funktion	Lesen	Schreiben	Aufw.
ARC1	ZIP vollstaendig: ZIP64 (>4 GB, >65535 Eintraege), AES-Verschluesselung (WinZip AE-1/AE-2) und das alte ZipCrypto, weitere Methoden im ZIP (BZIP2, LZMA, Zstd), Kommentare, Datei-Attribute und Zeitstempel	BCL: nur Deflate	BCL: nur Deflate	M
ARC2	BZIP2 (.bz2) — Burrows-Wheeler, Huffman	x	x	M
ARC3	XZ / LZMA / LZMA2 (.xz, .lzma) — der Bereichskodierer ist die Grundlage auch fuer 7z	x	x	L
ARC4	7z (.7z): Behaelter mit LZMA/LZMA2, BCJ-Filtern, fester Blockbildung („solid“), Verschluesselung	x	x	L
ARC5	Zstandard (.zst) — heute der Standard bei Linux-Paketen und Dateisystemen	x	x	L
ARC6	LZ4 (.lz4) — schnell, in Linux-Werkzeugketten verbreitet	x	x	S
ARC7	TAR-Ketten: .tar.gz, .tar.bz2, .tar.xz, .tar.zst als EINE Einheit behandeln, dazu die Unix-Rechte/Besitzer/Symlinks	TAR aus der BCL, ohne die Ketten		S
ARC8	RAR entpacken (RAR4 und RAR5): Behaelter, Huffman-/LZSS-Kern, Wiederherstellungsdaten, mehrteilige Archive	x	nie (siehe unten)	L
ARC9	CAB (Windows-Installationsarchiv, MSZIP/LZX) lesen und schreiben	x	x	M
ARC10	Apple: .dmg (HFS+/APFS-Abbild) und Apple Archive (.aar/.yaa) lesen	x	x	L
ARC11	ISO 9660 / UDF (Abbilddateien) lesen und schreiben	x	x	M
ARC12	Alte Mac-/Unix-Formate: Stuffit (.sit), compress (.Z), cpio, ar	x	x	M
ARC13	Fassade + MCP + Pruefstand: EIN Weg „Archiv oeffnen/erstellen“ ueber alle Formate, list_archive/extract_archive/create_archive, ein Reiter im Pruefstand	x	x	S

Ehrlich zu RAR: entpacken laesst sich nachbauen (das Format ist aus der frei verfuegbaren

unrar-Quelle erschliessbar). Packen nicht — die RAR-Lizenz untersagt ausdruecklich, aus diesen

Unterlagen einen RAR-Kompressor zu bauen. ARC8 ist deshalb bewusst nur „entpacken“; das ist keine

technische, sondern eine rechtliche Grenze, und sie bleibt bestehen.

Ehrlich zum Aufwand: ARC3/ARC4/ARC5 sind die dicken Brocken — ein LZMA- bzw. Zstd-Kompressor ist ein eigenes Stueck Handwerk (Bereichskodierer, Match-Suche, Woerterbuch). Entpacken ist jeweils deutlich leichter als packen; ein sinnvoller Zwischenstand waere „alles lesen, das Wichtige schreiben“.

v4-MUS · Notensatz — Notenblaetter erstellen, mit eigenem Editor (Nutzerwunsch)

Ziel: Notenblaetter mit allem, was dazugehoert, selbst setzen — und einen eigenen Editor dafuer.

Vorbilder sind Sibelius/Finale/Dorico und das freie MuseScore; der Austausch laeuft ueber MusicXML (das Format, das alle koennen) und MIDI.

Was dafuer schon da ist (nicht doppelt bauen): das Vektor-Zeichnen (PageCanvas), das Objekt-Modell mit Seiten (LayoutDocument) und sein .kdoc-Format, die Schrift-Einbettung samt Umriss-Auslesung und OpenType-Shaping (Notenschriften wie Bravura sind gewoehnliche OpenType-Schriften nach dem SMuFL-Standard), der abgesicherte XML-Leser (fuer MusicXML) und System.IO.Compression (fuer das gepackte .mxl), dazu die ganze Editor-Oberflaeche (Canvas, Auswahl, Undo/Redo, Paletten), auf der ein Noten-Editor aufsetzen kann.

#	Funktion	Stand bei dir	Aufw.
MUS1	Notenmodell: Partitur -> Stimmen -> Takte -> Ebenen -> Noten/Pausen; Tonhoehe, Dauer, Vorzeichen, Schluessel, Tonart, Taktart, Haltebogen, Balken, Triolen, Vorschlagsnoten		M
MUS2	Notenschrift ab Werk: eine SMuFL-Schrift buendeln (Bravura, SIL Open Font License — dieselbe Politik wie bei den Standard-14-Ersatzschriften M1100 und der Mathe-Schrift M1256) und ihre Glyphen ueber die vorhandene Einbettung setzen		S
MUS3	Stech-Regeln (der Kern): Notenkopf-Positionen, Halsrichtung und -laenge, Balkengruppen und ihre Neigung, Hilfslinien, Vorzeichen-Stapelung, Kollisionsvermeidung und die horizontale Verteilung (dauerabhaengiger Abstand)		XL
MUS4	Umbruch: Takte auf Systeme, Systeme auf Seiten, Zeilenausgleich, Wiederholungen und Sprungmarken (D.C./D.S./Coda), Taktzahlen		L
MUS5	Alles am Notentext: Dynamik, Artikulation, Bindeboegen, Triller/Verzierungen, Oktavierungen, Pedal, Fermaten, Atemzeichen, Tempo- und Vortragsangaben		L
MUS6	Text an den Noten: Gesangstext (mehrere Strophen, Silbentrennung, Melismen), Akkordsymbole, Fingersatz, Generalbass		M
MUS7	Mehrere Systeme: Klavier-Akkolade, Chor-/Orchesterpartitur, Systemgruppen und Klammern, Stimmenauszug aus der Partitur, Transposition (klingend notiert)		L
MUS8	Sonderfaelle: Schlagzeug-Notation, Gitarren-Tabulatur mit Griffbildern, Prozent-Wiederholungen, Cue-Noten, mehrtaktige Pausen		M

#	Funktion	Stand bei dir	Aufw.
MUS9	MusicXML lesen (.musicxml und gepackt .mxl) — der Weg herein aus jedem anderen Programm		L
MUS10	MusicXML schreiben — der Weg hinaus; damit ist der Rundlauf zu, wie bei DOCX/XLSX		M
MUS11	MIDI lesen und schreiben (Standard MIDI File 0/1): Noten aus einer Aufnahme uebernehmen bzw. die Partitur als MIDI ausgeben (Tempo, Anschlag, Programmwechsel)		M
MUS12	Als PDF setzen — ueber den vorhandenen PageCanvas/LayoutDocument-Weg, kein zweiter Zeichenweg; dazu SVG- und Bild-Ausgabe ueber die vorhandenen Wandler		M
MUS13	Der Noten-Editor (eigener Punkt, wie gewuenscht): Noten mit Maus und Tastatur eingeben, Ebenen, Auswahl und Verschieben, Paletten fuer Dynamik/Artikulation, Takt- und Systemoperationen, Undo/Redo — auf der vorhandenen Editor-Oberflaeche		XL
MUS14	Anhoeren: die Partitur als Klang wiedergeben (MIDI-Ausgabe an das System bzw. ein einfacher eigener Synthesizer fuer die Vorschau)		L
MUS15	MCP-Werkzeuge: musicxml_to_pdf, midi_to_pdf, pdf_to_musicxml-Ansatz sowie ein Pruefstand-Reiter		S

Ehrlich vorweg gesagt: MUS3 ist der eigentliche Brocken. Notensatz sieht nur dann gut aus, wenn Hunderte kleiner Regeln zusammenspielen (Abstaende nach Dauer, Halsrichtung nach Lage, Balkenneigung, Ausweichen bei Kollisionen) — das ist der Grund, warum es weltweit nur eine Handvoll ernsthafter Notensatz-Programme gibt. Der Rest der Liste ist Fleissarbeit, MUS3 ist Handwerk.

Zur Verifikation: fuer eingelesenes MusicXML ist music21 (Python) ein unabhaengiger Leser, und MuseScore/LilyPond koennen dasselbe Stueck zum Vergleich setzen. Nach der Projektregel geschieht das einmalig beim Bauen; im Repo bleibt die eingebackene Erwartung, kein Fremdwerkzeug-Aufruf.

v4-PRN · Virtueller PDF-Drucker (Nutzerwunsch)

Auf jedem Geraet „Drucken“ waehlen, das Ergebnis entsteht als PDF im Pruefstand und steht dort — mit Warteschlange — wieder zur Verfuegung. Bewusst KEIN Windows-Druckertreiber: der braeuchte ein signiertes Treiberpaket (WHQL/Code-Signing), einen Port-Monitor und liefe nur unter Windows. Der offene Weg ist ein IPP-Drucker (Internet Printing Protocol): Windows, macOS, iOS, Android und Linux drucken ohne jeden Treiber darauf, und moderne Clients senden den Job bereits als PDF.

#	Drucker-Funktion	Stand bei dir	Aufw
PRN1	IPP-Server (HTTP + Get-Printer-Attributes/Validate-Job/Print-Job/Get-Jobs/Cancel-Job): der Pruefstand meldet sich als Drucker „Kuonen PDF“ und nimmt den Job entgegen	Printing/lppPrintService (netzfreier Kern) + Playground/lppPrintServer (POST /ipp/print). Grenzen: keine Bekanntmachung im Netz (DNS-SD/AirPrint), kein IPP ueber TLS, keine Anmeldung; Print-Job laeuft gleichzeitig ab, daher greift Cancel-Job nur bei einem noch laufenden Auftrag	L
PRN2	Spooler + Oberfläche: Warteschlange mit Absender/Zeit/Seitenzahl/Status, Vorschau, Herunterladen, Loeschen, Aufbewahrungsdauer	fehlt	M
PRN3	Bekanntmachung im Netz (mDNS/Bonjour _ipp._tcp + AirPrint-Attribute) — der Drucker taucht von selbst in der Geraeteliste auf	fehlt	M
PRN4	Rueckgabe an das ausloesende Geraet: Link/QR-Code direkt nach dem Druck, wahlweise Ablage in einem ueberwachten Ordner oder Netzlaufwerk	fehlt (IPP selbst kann nichts zurueckgeben)	M
PRN5	PostScript-Port-Variante fuer Alt-Programme, die zwingend einen lokalen Drucker wollen: generischer Windows-PostScript-Treiber + eigene PPD -> Datei-Port -> der vorhandene PostScriptToPdf (M954). Kein eigener Treibercode, keine Signierung	fehlt	M
PRN6	Nicht-PDF-Jobs wandeln: PWG-Raster/URF (iOS/AirPrint) und PCL in PDF ueberfuehren	fehlt	L
PRN7	Zugang: Authentifizierung, IP-Beschaenkung, Groessen-/Mengenkontingent je Absender	fehlt (Bausteine aus AUTH1-3 da)	S
PRN8	Druckoptionen uebernehmen aus den IPP-Job-Attributen	M1327 (Printing/lppJobOptions) - Kopien und Ausrichtung werden angewandt, Duplex/Farbe/Format nur mitgefuehrt	M

v4-LR · Lightroom (Foto-Entwicklung und Katalog) — Nutzerwunsch

Die Bausteine dafuer sind zu grossen Teilen da: eigene Bild-Codecs (IMG), ICC-Farbmanagement mit echter Umrechnung (Icc), Resampler mit vier Verfahren (M1155), GPU-Kernel fuer Bildpunkt-Arbeit (Accel, M1023/M1031), Kontaktbogen (PhotoSheetComposer) und der PDF-/Druckweg. Was fehlt, ist die Foto-Entwicklung selbst und der Katalog.

#	Lightroom-Funktion	Stand bei dir	Aufw
LR1	RAW entwickeln: Demosaicing (Bayer/X-Trans), Weissabgleich als Temperatur/Tint, Kameraprofile	die Grundentwicklung steht mit M1309 (bilinear + Aufnahme-Weissabgleich); es fehlen X-Trans, Temperatur/Tint als Regler und echte Kameraprofile	XL
LR2	Grundeinstellungen: Belichtung, Kontrast, Lichter, Tiefen, Weiss, Schwarz, Klarheit, Dunst entfernen, Dynamik, Saettigung	fehlt	L
LR3	Tonwertkurve (Punkt + parametrisch je Kanal), HSL/Farbmischer, Farbkalibrierung	fehlt	L
LR4	Details: Schaerfen (Betrag/Radius/Details/Maskieren) und Rauschreduzierung (Luminanz + Farbe)	fehlt	L
LR5	Objektivkorrektur: Verzeichnung, chromatische Aberration, Vignettierung — rechnerisch und ueber Profile	fehlt	L
LR6	Transformieren: Perspektive/Ausrichten (Upright), Zuschnitt, Drehung, Spiegeln	fehlt	M
LR7	Lokale Anpassungen: Verlaufs- und Radialfilter, Pinsel, Bereichsreparatur — mit Masken	fehlt	XL
LR8	Effekte: Vignettierung nach Zuschnitt, Koernung	fehlt	S
LR9	Nicht-destruktiv: die Bearbeitung ist ein Rezept, das Original bleibt unberuehrt; Sicherung als XMP-Sidecar	fehlt (setzt IMG15 voraus)	M
LR10	Vorgaben (Presets) + Stapel-Synchronisierung ueber viele Bilder	fehlt	M
LR11	Katalog: Ordner/Sammlungen, Bewertung/Markierungen/Stichwoerter, Suche und Filter	fehlt (setzt IMG14/IMG15 voraus)	L
LR12	Export: Format/Groesse/Qualitaet, Ausgabeschaerfung, Wasserzeichen, Auswahl der mitgegebenen Metadaten	die Wandlung gibt es (convert_image, M1303), der Rest fehlt	M
LR13	Kontaktbogen / Web-Galerie / Fotobuch	Kontaktbogen da (PhotoSheetComposer), Galerie/Buch fehlen	M
LR14	GPU-Beschleunigung der Entwicklungs-Kette (die Kernel-Infrastruktur steht in Accel)	fehlt	L

Ehrliche Einordnung: LR1 und LR7 sind die beiden schweren Brocken; LR2/LR3/LR8 sind vergleichsweise geradlinige Bildpunkt-Arbeit auf dem vorhandenen Farb-Unterbau. Ein sinnvoller erster Schnitt waere

LR2 + LR3 + LR8 + LR9 — damit gibt es eine echte, nicht-destruktive Entwicklung fuer JPEG/TIFF/PNG, noch ohne RAW.

v4-EXT · Extern/Toolchain-gebunden (kein fehlender Code)

Aus v3 hierher verschoben — kein fehlender Code, sondern extern/toolchain-/urteilsgebunden.

Punkt	Stand	Was noch fehlt / warum
.wasm-Artefakt (browser-aufrufbar, kpdf_*-Exporte)	zurueckgestellt	JS-Loader + Drift-Waechter (M975) stehen; braucht NativeAOT-LLVM aus dotnet/runtime-Quelle + emscripten-SDK. Auf Nutzerwunsch aufgeschoben.
Native-Shim signieren (Authenticode/codesign)	CI-Schritt	Rezept dist/build-native-shim.sh (M1007) steht; es fehlt nur das Code-Signing-Zertifikat als CI-Secret.
PDF/UA-Semantik (Leseordnung/Alt-Text inhaltlich)	x menschlich	veraPDF prueft das Maschinell-Entscheidbare; ob ein Alt-Text die Abbildung inhaltlich beschreibt, bleibt menschliche Sichtpruefung.
Docker-Hub Untagged-Speicher automatisch freigeben	zurueckgestellt	Tag-Loeschung ist automatisiert (Cleanup Phase A, taeglich). Untagged-Digests belegen ~22 GB; belegt (3 gruene Cleanup-Laeufe lassen sie liegen): Docker Hub hat KEINE oeffentliche REST-API dafuer (Image-Management-UI = session-gebundener Loader). Einzige Automatik = ganzes Repo vor jedem Push loeschen — macht das Repo aber PRIVAT + verliert Beschreibung/Pull-Zaehler. Auf Nutzerwunsch aufgeschoben; bei Bedarf manuell ueber die Web-UI (Image Management) oder Docker-Hubs 6-Monats-GC.

v3 · Grosse Tracks — abgeschlossen

Gruppe	Worum es geht	Stand
V3-FM · Datei-Manager (nur MCP)	Ordner/Dateien lesen/schreiben/verschieben/kopieren/loeschen/umbenennen/anlegen + suchen (Name/Wildcard/Regex). Veraendernde Ops opt-in KUONEN_PDFTOOLS_ENABLE_FS_WRITE.	M1255
V3-AP · Apogee-Vorstufen-RIP-Kern	Halbton-Screening->Platte, Trapping, GCR/UCR, JDF-Workflow, Contract-Proof-Datei. Grenze: physische Proofer-Ansteuerung braucht das Hersteller-SDK.	M1269-1273
V3-W · LaTeX	eigene TeX-Engine W1–W28 (Feinheiten M1256–1266, u.a. \vfill).	M1266
V3-T · Barrierefreiheit / Validierung	maschinelle Scheibe im check_pdfua (M1268) + goldener Weg veraPDF (PDF/A + PDF/UA, MCP verapdf_validate, :tools-Image).	M1277
SEC-UI · Pruefstand-Seite „Sicherheit“	verschluesseln/entsperren/Passwort-Recovery in der UI.	M1232/1233
AUTH · Netzbetrieb-Authentifizierung	Basic-Auth (opt-in), Ausnahmeliste, ehrliche Fusszeile.	M1234

V3-FM · Datei-Manager - KOMPLETT (nur MCP; M1248/1249/1253/1255)

#	Was	Aufw.	Stand	Anleitung / baut auf
V3-FM	Datei LESEN/SCHREIBEN: read_file (Text, 5-MB-Deckel, Binaer benannt) + write_file (neu/anhaengen)	M	M1255	SystemFiles; write opt-in
V3-FM	VERSCHIEBEN/UMBENENNEN: move_path (Datei UND Ordner), overwrite	S	M1255	File.Move/Directory.Move gekapselt
V3-FM	DUPLIZIEREN: copy_path, rekursiv fuer Ordner, overwrite	S	M1255	File.Copy + Baumkopie
V3-FM	LOESCHEN: delete_path — Papierkorb (Vorgabe) ODER endgueltig (permanent=true), rekursiv	M	M1255	<Temp>/kuonen-papierkorb; Hard-Delete gated
V3-FM	Ordner ANLEGEN: create_directory	S	M1255	Directory.CreateDirectory
V3-FM	AUFLISTEN mit Modi: all/files/dirs/recursive	S	M1255	list_directory mode
V3-FM	SUCHE: Datei/Ordner/beides (type), Wildcard ODER Regex (regex), rekursiv/flach, optional Inhalt	M	M1255	search_files
V3-FM	AUSFUEHREN: run_process (Programm/.bat, inkl. Admin elevated M1253) + open_path	M	M1249/1253	opt-in KUONEN_PDFTOOLS_ENABLE_EXEC; Windows UAC/runas, Unix sudo
V3-FM	Gruppe dateisystem-schreib + Gating + Manifest/Taxonomie-Waechter	S	M1255	M1244-Sperrliste

V3-AP · Apogee-Vorstufen-RIP-Kern - KOMPLETT (M1269-1273)

#	Was	Aufw.	Stand	Anleitung
V3-A	Halbton-Screening -> CTP: Rasterweite/-winkel/Punktform (AM/FM) -> 1-Bit-Plattendaten je Separation; Winkelsatz C15/M75/Y0/K45, Dot-Gain-Kompensation	XL	M1269	Rip: HalftoneScreen/HalftoneScreener/DotGainCompensation/PlateRipper->TiffBilevel; MCP screen_plate. Grenze: gedrehte Rasterfunktion (kein Superzellen-Screening -> keine Moiré-Garantie fuer beliebige Kombis)
V3-A	Trapping: Ueberfuellung an Farbkanten (Trap-Breite, hell->dunkel, Schwarz-Aussparung)	L	M1270	Rip/TrapEngine auf CmykRaster. Grenze: feste rechteckige Trap-Breite, K nur ausgespart
V3-A	Tintenoptimierung GCR/UCR/TAC	L	M1271	Rip/InkOptimizer; Farberhalt der Neutralen ueber CIEDE2000 belegt. Grenze: 1:1-Grauaustausch
V3-A	JDF-Workflow / Job-Ticketing / Hot-Folder	L	M1272	Jdf: JdfReader/Writer, JmfMessage, JobQueue; MCP read_jdf. Grenze: JDF-Teilmenge (kein RIP)
V3-A	Contract-Proof-DATEI (CMYK-Simulation + Fogra-Medienkeil + Zertifizierungs-Label)	M	M1273	Proofing/ContractProof.Build; MCP contract_proof. Grenze: ein physisches Proofer-Geraet ANSTEUERN braucht das Hersteller-SDK

V3-W · LaTeX - KOMPLETT (W1-W28; Detail-Tabelle in v2-A)

#	Was	Aufw.	Stand	Anleitung
V3-W1	W4-Rest: \hfill/\dotfill/\hrulefill + vertikales \vfill	L	M1259/126	\vfill ueber Zwei-Pass, isoliert; v1: nur einseitig
V3-W2	W5-Rest: Knuth-Plass fuer LaTeX-Absaetze (einstilig + gemischt)	M	M1260/126	Optimal-Pfad M1235; gemischt ueber Konverter-Vorbruch
V3-W3+	2D-Mathe ab Werk: Mathe-OTF (TeX Gyre Termes Math, GUST) in Core gebuendelt als Standard-MathFont	M	M1256	keine Lizenzfrage; Muster wie StandardFontSubstitutes M1100
V3-W4	W20-Rest: echte Inline-Link-Annotationen	M	M1261	Nachbearbeitung PageTextExtractor->PdfLinkAnnotator
V3-W6	W22-Rest: automatische Silbentrennung (Knuth-Liang)	M	M1257	freie TeX-Muster de/en/fr in Core gebuendelt; babel schaltet ein

V3-T · Barrierefreiheit / Standard-Validierung - KOMPLETT (M1277)

#	Was	Stand
V3-T1a	Maschinell berechenbare Scheibe zusammenfassen + im check_pdfua sichtbar	M1268
V3-T1b	Goldener Weg: unabhaengige veraPDF-Validierung (PDF/A + PDF/UA), MCP + :tools-Image	M1277

SEC-UI · Pruefstand-Seite „Sicherheit" - FERTIG (M1232/1233)

#	Was	Aufw.	Stand
SEC1	Seite „Sicherheit": ENTFERNEN/ENTSCHLUESSELN (PdfUnprotector) + VERSCHLUESSELN (PdfProtector)	M	M1232
SEC2	Passwort-WIEDERHERSTELLUNG in der UI (Woerterbuch/Maske)	M	M1233
SEC3	Demo-Gating + Maschinen-Lizenz-Gate (security.removal) in der UI sichtbar	S	M1232

AUTH · Pruefstand-Authentifizierung - FERTIG (M1234)

#	Was	Aufw.	Stand
AUTH1	Basic-Auth-Middleware (Nutzer/Passwort aus Env, konstantzeit), opt-in, Vorgabe aus	M	M1234
AUTH2	Ausnahmeliste (Health/Herzschlag) + Reihenfolge Blockliste -> Auth -> Schranke	S	M1234
AUTH3	Fusszeilen-Hinweis ehrlich auf „fuer den Netzbetrieb gehaertet"	S	M1234

v2 · History der grossen Tracks (alle)

v2-A · Detail-Tabellen der grossen Tracks

W · LaTeX -> PDF (eigene TeX-Engine) - KOMPLETT (M1266)

#	Punkt	Aufw	Meilenst.	Anleitung / baut auf
W1	Tokenizer + Catcodes (Kommandos \..., Gruppen {}, %-Kommentare)	M	M1209	Latex/TeXTokenizer
W2	Makro-Expansion: \def/newcommand, Argumente #1..#9, \let, \expandafter	L	M1209	Latex/MacroExpander
W3	Zaehler + Laengen: \newcounter/\the/\value, \newlength, pt/em/ex/cm/in	M	M1216	Registertabelle im Konverter
W4	Box-/Leim-Modell: \hspace/\vspace/\mbox, \hfill/\dotfill/\hrulefill, \vfill	L	M1218/1259/1266	Leer-Inline-Kasten + Spacer; volles elastisches \hbox/\vbox-Leim = bewusste v1-Grenze
W5	Absatz-Umbruch (Knuth-Plass) ueber TextLayout	L	M1225/1235/1260/12	Algorithmus-Kern -> Live einstillig -> gemischt ueber Konverter-Vorbruch
W6	Seitenumbruch + \newpage/\clearpage/\pagebreak	M	M1209	FlowDocument.PageBreak
W7	Schrift-Auszeichnung \textbf/\textit/\emph, Groessen \tiny...\Huge	M	M1209	Standard-14-Abbildung
W8	Sonderzeichen/Ligaturen/~/\&/Gedankenstriche/Anfuehrungen	S	M1209	Zeichen-Mapping
W9	Mathematik inline $...$ + display $[...]$	M	M1209	LatexMathParser
W10	Mathe-Umgebungen equation/align/matrix/cases + &-Ausrichtung	L	M1263/1265	Matrizen+cases (M1263) + echte &-Ausrichtung (M1265), ab Werk via M1256
W11	Listen itemize/enumerate/description + \item	M	M1209	FlowDocument-Listen
W12	Tabellen tabular/array (l c r p{} , \hline, &, \)	L	M1210	TableFrame
W13	\multicolumn/\multirow/booktabs	M	M1210	TableFrame-Merge
W14	Gleitobjekte figure/table + \caption	M	M1213	FlowDocument-Floats (v1 im Fluss)
W15	\includegraphics (graphicx) — PNG/JPEG einbetten	S	M1211	ImagePlacement
W16	Querverweise \label/\ref/\eqref (Zwei-Pass)	M	M1212	Fixpunkt (\pageref offen)
W17	Bibliographie thebibliography/\cite	M	M1212	nummerierte Liste + [n]
W18	Fuss-/Randnoten \footnote	S	M1215	ParagraphWithFootnotes + BuildWithFootnotes
W19	Abschnitte + Nummerierung + \tableofcontents	M	M1209	Outline + BuildWithTableOfContents
W20	hyperref: \href/\url + Lesezeichen + Inline-Links	M	M1217/1261	Block-Link + Inline-Annotationen

#	Punkt	Aufw	Meilenst.	Anleitung / baut auf
W21	geometry: Seitengroesse/Raender/Orientierung	S	M1211	PageSize/Margins
W22	babel: Sprache + Silbentrennung	M	M1219/1257	deutsche "-Kurzbefehle + Trennmuster de/en/fr
W23	Kopf-/Fusszeilen fancyhdr/thepage	S	M1214	HeaderFooterStamper
W24	Farbe xcolor: \textcolor/definecolor	S	M1209	Farb-Setter
W25	Dokumentklassen + \maketitle	M	M1209	Praeambel-Auswertung
W26	\input\include (lokale Dateien, Tiefendeckel)	S	M1211	sichere Datei-Einbindung
W27	Fehlertoleranz + LatexConversionReport	M	M1209	Report wie HTML
W28	Anbindung LatexConverter.Convert + MCP latex_to_pdf	M	M1209	Fassade

XF · XSL-FO / Apache-FOP-kompatibel -> PDF - KOMPLETT (M1193 + M1198)

#	Punkt	Aufw.	Meilenst.	Anleitung / baut auf
XF1	<fo:root>/layout-master-set/simple-page-master (Seitengeometrie, Raender)	M	M1193	Fo/FoToPdf + PageSize
XF2	page-sequence/flow/static-content (Kopf-/Fusszeilen)	M	M1193	FlowDocument + HeaderFooterStamper
XF3	<fo:block> (text-align/text-indent/space-after) + <fo:inline>	M	M1193	TextStyle/StyledRun
XF4	<fo:table> - Spans	L	M1193	TableFrame (Spans v1-Grenze)
XF5	list-block/list-item, external-graphic/instream-foreign-object	M	M1193	List/Image/Svg
XF6	page-number/page-number-citation, footnote; leader	M	M1193	{page} im Stamper
XF7	basic-link; marker/retrieve-marker	M	M1193	Link (Marker v1-Grenze)
XF8	Bidi (writing-mode/bidi-override), Mehrspalten (column-count)	M	M1198	BidiAlgorithm + MultiColumnRegion
XF9	volles FO-Property-Set (Vererbung, inherit, from-parent) + Report	L	M1198	volle Property-Kaskade
XF10	XSLT (XML + .xsl -> XSL-FO) ueber System.Xml.Xsl	M	M1193	FoToPdf.ConvertXml
XF11	Anbindung FoToPdf.Convert + MCP fo_to_pdf + Beispiel	M	M1193	XslFoBibliothek

PPTX · PowerPoint lesen/schreiben - KOMPLETT (M1205)

#	Punkt	Aufw.	Meilenst.	Anleitung / baut auf
PX1	PPTX-Leser (PptxReader): slideN.xml -> Folien mit positionierten Text-/Bild-Rahmen	L	M1205	ZIP+XML wie DocxReader
PX2	PPTX -> PDF (PptxToPdf) + MCP pptx_to_pdf	M	M1205	PdfDocumentBuilder/PageCanvas
PX3	PPTX-Schreiber (PptxWriter) + PdfToPptx + MCP pdf_to_pptx	L	M1205	ZIP+XML wie DocxWriter

XFA-fließend · dynamisches XFA - KOMPLETT (M1190 + M1198)

#	Punkt	Aufw.	Meilenst.	Anleitung / baut auf
XR1	fließendes Subform-Layout (layout="tb")	L	M1198	Xfa/XfaRenderer
XR2	datengesteuerte Zeilen-Vervielfachung (occur) + mehrseitig	M	M1198	Datenbindung je Wiederholung
XR3	echte <ui>-Darstellung + Rahmen-/Fuellstile	M	M1198	Appearances-Muster

X · PDF/PS -> DOCX (Rekonstruktion) - VOLLSTAENDIG (X1-X15)

#	Punkt	Meilenst.
X1–X8	OOXML-Writer, Absätze/Laeufe/Tabellen/Bilder, Leseordnung	M1073/1164
X9	getaggttes PDF -> Formatvorlagen (Struktur-Baum)	M1184
X12	Kopf-/Fusszeilen aus wiederholtem Rand-Text	M1185
X13	PS->DOCX (PostScriptToDocx) + MCP pdf_to_docx/ps_to_docx	M1164

ED14-voll · PDF -> editierbares LayoutDocument (Acrobat-Kern) - KOMPLETT (M1062-1071)

#	Punkt	Aufw.	Meilenst.	Anleitung / baut auf
ED14-2	Vektorpfade -> ShapeFrame	L	M1062	DisplayListBuilder-Pfade
ED14-3	AcroForm-Felder -> FormFieldFrame	M	M1067	FormFieldEditor M965
ED14-4	Annotationen -> LayoutAnnotation	M	M1068	AnnotationEditor M962
ED14-5	Font-Uebernahme (eingebettete Fonts statt Standard-14)	L	M1063	FontExporter M990

#	Punkt	Aufw.	Meilenst.	Anleitung / baut auf
ED14-6	Absatz-Rekonstruktion	M	M1064	wie X3
ED14-7	Bild-Objekte mit SMask	S	M1065	ImageExporter
ED14-8	Tabellen-Erkennung -> TableFrame	M	M1066	PdfTableExtractor
ED14-9	Ebenen/OCG -> LayoutDocument.Layers	M	M1069	ED8 M1039
ED14-10	.kdoc-Rundlauf fuer alle Typen	M	M1070	KdocConverter
ED14-11	MCP pdf_to_kdoc + Beispiel	S	M1071	Fassade

IMG2-Pixel · HEVC-Intra-Rekonstruktion - KOMPLETT (M1111-1147, PixelsDecoded=true)

#	Punkt	Aufw.	Meilenst.	Anleitung
IMG2-1	Slice-Segment-Header	M	M1111	ueber die CABAC-Maschine M1006
IMG2-2	CTU-Quadbaum + Residuen	L	M1118-1124	CABAC-Parse gegen libde265-Trace
IMG2-3	Intra-Praediktion (alle 35 Modi)	XL	M1114	ISO 23008-2
IMG2-4	Transform-Baum + inverse DCT/DST	L	M1112	eigener IDCT/IDST
IMG2-5	Dequantisierung	M	M1113	§8.6.3
IMG2-6	Residuen + Praediktion zusammenfuehren	M	M1115/1126	AddResidual
IMG2-7	In-Loop-Filter (Deblocking + SAO)	L	M1116/1117	Luma/Chroma
IMG2-8	YUV->RGB + Anbindung images_to_pdf	S	M1147	Chroma 4:2:0 + BT.601

PUB · PDF -> Publishing-/Office-Format - KOMPLETT (M1125/1128/1071)

#	Punkt	Aufw.	Meilenst.	Anleitung
PUB1-3	ODF .odt (Geruest + Textlaeufer/Absaetze/Tabellen/Bilder/Listen)	M	M1125	Odf/OdfWriter/PdfToOdf, Orakel odftp
PUB4	.odg-Zeichnung (frei positionierte Rahmen)	L	M1128	Odf/OdgWriter/PdfToOdg
PUB5	MCP pdf_to_odf/pdf_to_odg + Beispiel	S	M1128	Fassade
PUB6	Alternativ: PDF -> .kdoc (ED14-voll, MCP pdf_to_kdoc)	—	M1071	= ED14-voll

Kleinere Einzelpunkte - KOMPLETT (M1051-1150)

#	Punkt	Meilenst.
R3-Rest	bare-CFF + Type1-Glyphen zu Pixeln	M1059 (+ Std-14-Ersatz M1100)
R4-Rest	Mesh-Shadings (Typ 4–7) zu Pixeln	M1057
P1-Rest	Type42/TrueType im PostScript	M1060
D2-Rest	XFDF-Annotationen lesen/schreiben	M1052
UA5-Rest	weitere Matterhorn-Pruefpunkte	M1000/1019/1051
B13a/b/c	MaxiCode / Han-Xin / DotCode	M1054/1061/1053 (gegen bwip-js/zint)
B15	Klarschrift/HRI in der PNG-Ausgabe	M1055
FL-Rest	Absatz-weiter Bidi + Float-Umfließen je Spalte	M1022/1056
G6-Ausbau	Stroke/Text/Bild/Shading/Clip auf der GPU	M1058/1142/1150

v2-B · Externe/Toolchain/Lizenz-Grenzen (kein Code fehlt)

Punkt	Stand	Warum
Fogra/Ugra-L*a\b\-\Sollwerte	- eingebaut M1186	freie zertifizierte FOGRA51/52 als Ressource (ICC-Registry), ueber CgatsReader. Physischer Medienkeil/Altona = kostenpflichtige Produkte, nicht nachgebaut.
Authenticode/codesign (Native-Shim)	CI-Secret	Rezept dist/build-native-shim.sh (M1007) steht; nur das Zertifikat fehlt.
.wasm-Artefakt (kpdf_*-Exporte)	zurueckgestellt	braucht NativeAOT-LLVM aus Quelle + emscripten (kein dotnet publish). JS-Loader + Drift-Waechter (M975) stehen.
PDF/UA-Semantik (Leseordnung/Alt-Text inhaltlich)	x menschlich	veraPDF (M1277) prueft das Maschinell-Entscheidbare; Bildverstaendnis bleibt menschliche Sichtpruefung.
WCAG-Kontrast	- maschinell geprueft	WcagContrast + ContrastAnalyzer (WCAG-2.1, AA/AAA), wo Farben extrahierbar.

v2-C · Nie im PDF (kein Gegenstueck)

Punkt	Status
Dynamisches XFA rendern (Template->Seiten)	x Adobe-proprietäer, seit PDF 2.0 deprecated. Daten BEFUELLEN ist da (XfaDataFiller M1028).
HTML-Interaktivitaet (JS-Events, :hover, Live-Verhalten)	x kein Laufzeit-Gegenstueck. Lade-Zeit-<script>+DOM+Reflow als opt-in Add-on (M1026).

Punkt	Status
HTML-Animation/Medien (Transitions, Video/Audio, WebGL, iframes, sticky)	x Standbild/erster Frame ist das Aeusserste.
@media Laufzeit (orientation, prefers-color-scheme)	x keine Laufzeit-Umgebung. BREITEN-Merkmale werden ausgewertet (M651).

v1 · Archiv — der abgeschlossene Stand je Bereich

ALLE Bereiche A-AN sind KOMPLETT. Genuin offen bleiben nur die extern-/toolchain-/urteilsgebundenen Punkte der v2-B-Tabelle.

A · PDF-Rasterizer (Render-Engine) - KOMPLETT (R1-R6, M960-993)

#	Punkt	Aufw.	Status
R1	Content-Stream -> Display-List (Grafikzustand/CTM/Pfade/Text/Bild-Do/Clip)	L	M960
R2	Pfad-Rasterung: Fill (nonzero/even-odd) + Stroke + Anti-Aliasing	L	M963
R3	Glyphen: eingebettete TrueType/CFF/Type1 + Std-14-Ersatz -> Pixel	L	M966/1059/1100
R4	Bilder (Bild-XObjects) + axial/radial + Mesh-Shadings	M	M969/976/1057
R5	Mischmodi (trennbar + nicht-trennbar) + Overprint + weiche Masken	M	M970/982/993
R6	PdfRenderer.RenderPage(dpi) + Thumbnails + Pixel-Diff + MCP render_page	S	M973

B · PDF-Format (ISO 32000) - KOMPLETT

#	Punkt	Status
v3-1	JBIG2-Dekoder (/JBIG2Decode)	M677-700
v3-2	JPEG-2000-Dekoder (/JPXDecode)	M702-738
v3-3	Linearisierung (Fast Web View)	M876
v3-4	PDF/A-Erzeugung (A-1/A-2/A-3)	M719-742
v3-5	PDF/UA-Konformitaet	M866
v3-5b	PDF/X-5 / PDF/X-6	M875/878/882

#	Punkt	Status
v3-6	Multimedia-Annotationen	M873
v3-7	Geospatial/Measurement PDF	M872
v3-8	Collections/Portfolios	M871
v3-9	Seitenuebergaenge/Artikel-Threads/Thumbnails	M869-870
v3-10	Type3-Fonts erzeugen	M874
v3-11	Lab/CalRGB/CalGray erzeugen	M868
v3-12	cmap Format 2 (gemischt CJK)	M814

C · HTML/CSS -> PDF (v2) - KOMPLETT (M646-951)

#	Punkt	Status
v2-1	Box um beliebigen Block-Inhalt	M889
v2-2	Hintergrund-Feinheiten (size/position/repeat, data:, SVG-Hg)	M890/891/918/930/931
v2-3	border-radius vollstaendig	M896/928/932
v2-4	Verlaeufer (Laengen-Stopps, Winkel, conic)	M653/797/934/935
v2-5	Relative Masse %/vw/vh/auto + Hoehe ueber Seitenumbrueche	M887/898/926/943
v2-6	Feinsatz auf GEMISCHTEN Laeufen	M927
v2-7	vertical-align gegen Zeilenboxen-Maximum	M914/939
v2-8	position-Rest (rel-Container, Inline-span-absolute)	M666/672/933/951
v2-9	transform-Rest (%-translate, transform-origin, Inline)	M671/886
v2-10	CSS-Sammelposten (text-shadow/filter/blend/writing-mode/columns/clip-path/outline/object-fit)	M899/900/917/921/929/936/937/948/950
v2-B	Teil B: box-sizing/calc/var/Print-CSS/::before-after/counter/Flex/Grid/Selektoren/@font-face/Tabellen-CSS	M883-941

D · HTML-Formular (v2-11) - KOMPLETT

#	Punkt	Status
F1	Feldgroesse aus CSS/Attributen	M901
F2	Attribute -> AcroForm-Flags (required/readonly/maxlength/title)	M893/924
F3	Feld-Stil aus CSS (Rahmen/Hintergrund/Schrift)	M902
F4	input-types (email/number/tel/url/date -> Textfeld)	M894
F5	Vorgabewert (input value + option selected)	M892
F6	label for / fieldset+legend	M925

E · Barcodes erzeugen - KOMPLETT

#	Punkt	Status
B1–B4	Opazitaet / Verlaeuft / Logo / Mittel-Symbol	M743/745/759/760
B5	QR-Modulform (runde Module)	M750
B6–B7	ISBN / EAN-Add-ons	M744/746
B8–B9	DataMatrix / Aztec	M753-757
B10	Weitere 1D (ITF/Codabar/Code93/GS1 DataBar/MSI/Pharmacode ...)	M747-781
B11	Micro QR / rMQR	M762/778
B12	QR Structured Append	M761
B13	MaxiCode/Han Xin/DotCode	M1053/1054/1061
B14	Klarschrift ueber UND unter	M748
B15	Klarschrift in PNG-Ausgabe	M1055

AM · Barcodes LESEN - KOMPLETT

#	Punkt	Status
BR1	Vorverarbeitung (Graustufen/Sauvola/Deskew/Entzerrung)	M769/782-785/811

#	Punkt	Status
BR2	1D-Leser (EAN/UPC, Code128/39/93/ITF/Codabar/MSI)	M763-765/772
BR3	QR-Leser (Finder/RS-EC/Demaskierung/Perspektive)	M767/782/783
BR4	PDF417-Leser	M774
BR5	DataMatrix + Aztec-Leser	M768/773/784
BR6	Mehrere Codes je Bild	M777/792
BR7	GPU-Beschleunigung	M785/897
BR8	MCP read_barcodes + Playground (ROI-Editor)	M771/802/861-863

Bereiche F-AL - alle KOMPLETT

Bereich	Unterpunkte (alle)	Stand
F · SVG	S1 Verlauf M319/320 · S2 Gruppen-Deckkraft/Maske M317/318 · S3 foreignObject->HTML M995 · S4 textPath M322 · S5 SMIL-Standbild M980 · S6 Selektoren+Units M995	M995
G · PostScript	P1 eingeb. Fonts (Type3 M981/Type1 M994/Type42 M1060/CID M1187) · P2 Farbraeume M981 · P3 shfill+Tiling M310/311/315 · P4 Text-Rest M981 · P5 Bild/Daten M981	M1187
H · EPS + Anbindung	E1 DOS-EPS M808 · E2 platzierbar (EpsFrame/FlowDocument.Eps) M954 · E3 MCP ps_to_pdf/eps_to_pdf M954	M954
I · Daten & Formulare	D1 FDF · D2 XFDF+Annots M1052 · D3 Form->JSON/CSV · D4 Tabellen-Extraktion · D5 XFA fuellen M1028 + rendern M1190 · D6 inkrement. Fuellen	M1190
J · Fluss-Layout	FL1 Fussnoten · FL2 Witwen/Waisen · FL3 RTL/vertikal · FL4 Zeilenr. · FL5 TOC/Index · FL6 Querverweise · FL7 Seiten-Floats; Absatz-Bidi+Spalten-Float M1056	M1056
K · Objektmodell	OM1 Charts · OM2 Text-auf-Pfad · OM3 bedingter Inhalt	M986
L · Text/Schriften	TX1 vertikal CJK · TX2 Thai/Tamil/Bengali/Khmer/Myanmar M1029 · TX3 Kapitaelchen	M1029
M · Absatz-/Zeichenformate	ST1 Einzuege · ST2 Case · ST3 Tracking/Unterstr. · ST4 keep-with-next · ST5 Silbentr. · ST6 Nummerierung · ST7 Absatzlinien	M978
N · GPU-Beschleunigung	G1 Bild-Infra · G2 Resample · G3 SVG-Filter · G4 Farbtransform · G5 JPEG-IDCT · G6 Rasterizer-GPU (Fill/Stroke/Clip/Bild/Blend/Shading/Text) · G7 OCR-Vorverarb. · G8 Durchsatz	M1150
O · Farben & Verlaeufer	C1 Tiling-Pattern M956 · C2 Lab/CalRGB/CalGray M868 · C3 ShadingType1 M956 · C4 SMask-API M956 · C5 Pattern-Farbraum M956	M956
P · Druckvorstufe	DV1 Falzschema · DV2 Creep · DV3 Schoen/Widerdruck · DV4 Rollendruck · DV5 Ink-Vorschau · DV6 Messstreifen	M983/999

Bereich	Unterpunkte (alle)	Stand
Q · Farbmanagement	CM1 Bild/XObject-Konv. M984/1001 · CM2 Papierweiss · CM3 Standardprofile · CM4 .icc-Export · CM5 Farb-Setup · CM6 MCP export/list_color_profiles	M1001
R · Sicherheit	SEC1 Public-Key-/Zertifikat-Verschlüsselung (PubSec, /Adobe.PubSec, AES-256) M957	M957
S · Signaturen	SIG1 RSA-PSS M256 · SIG2 PAdES-Stufen M955 · SIG3 Trust-Anker + /VRI M955	M955
T · Barrierefreiheit	UA1 Kennung M866 · UA2 Prüfer M866 · UA3 /Tabs M979 · UA4 Abk.-Expansion M979 · UA5 Grundpflichten M1051 + Kontrast/Leseordnung/Alt-Text-Audit M1268 · veraPDF M1277	M1277
U · Bildformate	IMG1 WebP M977/1003 · IMG2 HEIF/HEIC-Pixel M1111–1147 · IMG3 GIF M977	M1147
V · Add-on-DLLs	Z1 Netz-Ressourcen · Z2 AcroForm-JS · Z3 Dokument-JS · Z4 Browser-Engine M1026 · Z5 TS->JS M1020	M1026
W · LaTeX -> PDF	eigene TeX-Engine W1–W28 (Detail-Tabelle in v2-A)	M1266
X · PDF/PS -> DOCX	X1–X15 (X9 getaggt M1184 · X12 Kopf-/Fusszeilen M1185 · MCP pdf_to_docx M1073)	M1185
Y · HTML -> DOCX	HD1 DOCX-Emitter · HD2 E2E · HD3 MCP html_to_docx	M1014
Z · Tabellen -> XLSX	XLSX-Writer, Zell-Datentypen, HTML/PDF-Tabellen -> Register, MCP html_to_xlsx/pdf_to_xlsx	M1030
AA · Office-Import	DI1–3 DOCX-Leser M1012 · XI1–3 XLSX-Leser M1013	M1013
AB · Font-Export aus PDF	FX1 Audit M824 · FX2 ttf/otf · FX3 bare-CFF->OTF · FX4 Type1->PFB/PFA · FX5 cmap aus /ToUnicode · FX6 MCP · FX7 FontHealthReport M1002 (FX2-6 M990)	M1002
AC · Font ersetzen/reparieren	FR1 Reparieren M952 · FR2 Prüfer M953 · FR3 andere Schrift M953 · FR4 MCP M953	M953
AD · Signaturen exportieren	SG1 CMS/.p7s · SG2 Zert./Kette · SG3 Zeitstempel+DSS · SG4 Bericht · SG5 MCP	M989
AE · Externes Signieren	DS1 vorbereiten · DS2 CMS einpflanzen · DS3 MCP prepare/embed_signature	M989
AF · Passwort-Recovery	PR1 Prüfer · PR2 Woerterbuch · PR3 Maske/Brute · PR4 RC4-40 · PR5 GPU M1011 · PR6 MCP recover_password	M1011
AG · Sprachbindungen	IB1 C-ABI M958 · IB2 Header M961 · IB3 Python M964 · IB4 Node M967 · IB5 Java M972 · IB6 WASM-Loader M975 · IB7 REST+gRPC M1004/1025 · IB8 CLI M1004 · IB9 Go/Rust/PHP/Ruby/Swift M1005 (browser-.wasm v2-B)	M1025
AH · Geschuetzte Auslieferung	IB10 Schutz-Pipeline alle Assemblies M910/911 · IB11 native Shim M1007	M1007
AI · Editor InDesign	ED1–ED13 + ED26 + LF1–LF4 (WYSIWYG/Inline-Text/Undo/Mehrseiten/Hilfslinien/Paletten/Blend/Schatten)	M1032-1039
AJ · Editor Acrobat	ED14-voll PDF->Layout M1062-1071 · ED15-ED19 M962-974 · ED20 Lesezeichen/Links M1009	M1071
AK · .kdoc-Ausbau	ED21–ED24 M1009 · KG1 Spezial-Objektarten M1149–1152 · KG2 Fonts als Textschrift M1144 · KG3 Named-Styles M1145	M1152
AL · REST-Paritaet	ED25 REST-Host M1010 · Cover/Einband M1159; Editor=REST=Library-Paritaet ist stehende Regel	M1159

AN · NEU dazugekommen (nicht in v13) - KOMPLETT

#	Track	Meilenst.
N1	OCR-Ausbau A–F (CRNN-Destillation, Woerterbuch+n-gram, Layout, Deskew, GPU-CRNN, Unicode-Textebene, Sprachen)	M600-629
N2	Gerichtsakten/Documents (Fallnummer, organize_cases, describe_documents, search_case_number, Gerichtscodes)	M630-640
N3	MCP-Vollabdeckung (~40 Werkzeuge: watermark/encrypt/optimize/sanitize/bookmarks/redact/sign/generate_pdf/...)	M641/642+
N4	PDF/A-Lebenszyklus (Ordner-Konvertierung, Bereitschaftsbericht, Triage, Fremd-PDF->PDF/A + Schrift-Nacheinbettung)	M816-832
N5	Pruefstand-Werkzeuge (Merge/Split/Bild->PDF/Optimieren, „Barcode lesen“-Seite mit ROI-Editor, PDF/A-Reiter)	M806-863
N6	Infrastruktur (MCP-Auto-Update + Selbst-Tausch, Docker-Hub-Publishing, geschuetztes Zwei-DLL-ZIP)	M852-913