

Roadmap — erledigt in v4

Farb-Legende (Status):

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

Teil der Roadmap — dort steht, was noch offen ist.

MG · Gemeldete Maengel an bereits Gebautem

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 Zurueckblaettern. Jetzt ein Speicher je Seite (RenderCache, testbar) und 96 dpi	M	
MG5 · Import: Text kommt als Kaestchen an	Eine echte Rechnung (KPT) zeigte im Editor statt Text nur eine Reihe kleiner dunkler Kaestchen	Erledigt (M1373). Am Dokument gemessen: der Erzeuger setzt /F2 1 Tf und traegt die Groesse in die Textmatrix (10 0 0 10 x y Tm) — der Import nahm nur die Tf-Zahl und schrieb fontSize: 1. Weil die Absatz-Buendelung (M1314) ihre Grenzen in Vielfachen der Groesse misst, zerfiel damit jede Zeile in Ein-Zeichen-Objekte (799 Objekte auf Seite 1). Neu TextRun.EffectiveFontSize = Laenge des transformierten y-Einheitsvektors der Renderingmatrix (gilt auch gedreht/geschert), im Import ueber VisibleFontSize genutzt. Nachher: 43 Objekte, Groessen 6–14 pt	S	

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.	XL	Accessibility/AutoTagger repariert fremde PDFs auf Byte-Ebene, MCP auto_tag_pdf. Das Reading-Order-Werkzeug ist M1373 (ED41): Editing/StructureTreeEditor (Rolle/Alternativtext/Kinder-Reihenfolge eines vorhandenen /StructTreeRoot an Ort und Stelle aendern) + Playground/Model/AccessibilityTreeView (Baum->flache Zeilen) + der Reiter „Tag-Baum / Leseordnung“ im Editor-Overlay
AP2 · Prepare Form (Feld-Autoerkennung)	Formularfelder auf einem flachen PDF/Scan automatisch erkennen + anklickbar anlegen (Text/Checkbox/Radio).	L	Extraction/FormFieldDetector (erkennen) + FormComposition/FormFieldAutoPlacer (anlegen ueber den vorhandenen Formular-Weg), MCP detect_form_fields. Was nicht sicher zuzuordnen ist, wird benannt statt geraten. Grenzen: reiner Bild-Scan braechte Bilderkennung; Optionsgruppen und Auswahllisten sind aus der Geometrie allein nicht erschliessbar
AP3 · Bild in-place bearbeiten	Ein Bild in einem fremden PDF anklicken -> verschieben/skalieren/ersetzen, ohne Import-Rundlauf.	L	Editing/PdfImageEditor (auflisten/ersetzen/verschieben/loeschen) + MCP edit_image. Grenzen: nur unmittelbar auf der Seite gezeichnete Bilder - Form-XObject und Inline-Bild werden benannt abgelehnt; MoveImage erkennt die einfache q cm Do Q-Form
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	PdfRichMediaAnnotator + Pdf3DAnnotator + AnnotationPosterAppearance, MCP add_rich_media/add_3d_annotation. Ehrlich: die Bibliothek spielt nichts ab und rendert kein 3D - sie bettet ein; /RichMedia ist eine Adobe-Erweiterung ausserhalb von ISO 32000, in PDF/A verboten
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).	XL	M1373 (ED42): Playground/Components/Editor/ForeignPdfEditorPanel - dasselbe InPlaceEditor (kein zweiter In-Place-Weg) jetzt auch als Overlay der grossen Editor-Seite (Menue Datei -> „Fremdes PDF an Ort und Stelle bearbeiten...“), mit dem Reiter „Tag-Baum/Leseordnung“ (ED41) daneben. Grenzen: kein Reflow, keine Form-XObjects (unveraendert, aus InPlaceEditor)

v4-LID · InDesign

#	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 Naehierung, 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: Skalieren, Neu zentrieren, Objektbasiert) + M1430 die hilfslinienbasierte Regel. Dafuer zuerst die Bibliotheksfaehigkeit: LayoutGuide + LayoutPage.Guides (Ausrichtung, Lage, Name, Kennzeichen „fluessig“), im .kdoc nur geschrieben, wenn vorhanden, nie gezeichnet. Die Regel folgt einem Nahtmodell (nicht Adobes Algorithmus, das steht so in der Klassendoku): jede fluessige Linie nimmt ihren gleichen Anteil der Groessenaenderung auf, ein durchschnittenes Objekt waechst, ohne fluessige Linie auf einer Achse gilt Neu zentrieren; eine Linie genau auf einer Objektkante schneidet nicht. LiquidLayout.ApplyToDocument erhaelt Verkettungen; MCP liquid_layout; im Editor sind die Lineal-Hilfslinien jetzt je Seite im Modell (vorher global und nach dem Neuladen weg), als fluessig markierbar und das ganze Dokument per Liquid Layout umrechenbar. Gemeldet statt still: Doppelseiten, Anmerkungen und Musterseiten werden nicht umgerechnet; gleiche Anteile je Naht (keine Gewichtung), Schrumpfen kappt bei 0, Objektdrehung fliesst nicht ein	L

v4-LWD · Word (.docx)

#	Word-Funktion	Stand bei dir	Aufw
LWD	Aenderungsverfolgung LESEN: Accept-Changes-Endfassung (w:ins/w:del) M1289 + Markup-Ansicht (Einfuegung gruен, Loeschung rot durchgestrichen) + Kommentare (word/comments.xml in den Lesebericht) - ehrlich benannte Grenze: Aenderungsverfolgung und Kommentare schreiben nicht umgesetzt	M1289+	L
LWD	Fussnoten in .docx LESEN (w:footnoteReference) M1286 + SCHREIBEN (DocxWriter legt word/footnotes.xml an) + Endnoten lesen (ans Dokumentende, eigener Zaehler)	M1286+	M

#	Word-Funktion	Stand bei dir	Aufw
LWD:	Felder: w:fldSimple M1288 + komplexe Felder (w:fldChar Begin/Separate/End) - je der zwischengespeicherte Wert, nicht der Feldcode; die Felder werden nicht neu berechnet (ehrlich benannte Grenze)	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 (lesen, ueber den vorhandenen Formelsatz gesetzt) + Docx/LatexToOmml (schreiben, M1356-Rest): Brueche, Wurzeln (auch mit Grad), Hoch-/Tiefstellung, mitwachsende Klammern, grosse Operatoren mit Grenzen (m:nary, Summe/Produkt/Integral - undOvr bzw. subSup je nach \int gegen \sum) und griechische Buchstaben/Symbole als Roh-Unicode. Ohne Mathe-Schrift bzw. bei Akzenten/Ueber-Unterstrichen/Matrizen wird die Formel gemeldet statt geraten (kein Halbergebnis)	L
LWD:	Textrahmen/Formen/SmartArt/Diagramme	M1356 (Textrahmen als Absatz) + M1431: neues gemeinsames Modul DrawingML (Vorlage- und freie Geometrie, Fuellung, Kontur, Themenfarben samt lumMod/lumOff/tint/shade im linearen Raum, Transformation, Textkoerper, Diagrammleser) - PowerPoint und Excel lesen darueber, kein dritter Parser. Word-Formen, Textrahmen (Rahmen + formatierter Text, Innenabstand, senkrechte Ausrichtung) und Gruppen werden gezeichnet; wp:inline fliesst im Text, wp:anchor sitzt absolut nach positionH/positionV (page/margin/column/paragraph, posOffset/align/pctPos), Umfluss ueber FlowDocument.FloatObject. SmartArt ueber die von Word abgelegte vorgerenderte Zeichnung (word/diagrams/drawingN.xml) - kein nachgebauter Layout-Algorithmus; ohne Zeichnung die Knotentexte als Liste nach der Hierarchie (DOCX_DIAGRAM_TEXT_ONLY). Diagramme als natives ChartFrame. Sollwerte aus Microsoft Word 16 (Fixtures per COM erzeugt, PDF-Export vermessen) und LibreOffice. Satzspiegel aus w:pgMar statt fest. M1432 (die M1431-Grenzen geschlossen): alle 187 Vorlageformen ueber einen Formelrechner nach ECMA-376 §20.1.9 samt Textrechteck, freie Konturen mit mehreren Pfaden; Text im Textrechteck mit Absatzformat, Aufzaehlung, normAutofit/spAutoFit, senkrechter Schreibrichtung, Tabellen und Bildern; Konturen mit Deckkraft, allen Strichmustern, Pfeilspitzen, Mehrfachlinie; Fuellungen mit 54 Mustern, Bild, Pfad- und mehrstufigem Verlauf; Schatten/Schein/weiche Kante/Spiegelung/Relief silhouettentreu; verankerte Formen hinter dem Text, Umfluss eng und mehrspaltig; alle Diagrammartentypen ausser Oberflaechen samt Achsentitel, Datenbeschriftungen, Gitter, Achsengrenzen, Log-Achse, Zahlenformaten; .docx direkt nach .kdoc (DocxToKdoc, Schema §8.21/§8.22); fette Standardschrift mit richtigem Vorschub im Rasterizer. Orakel Word 16 und PowerPoint 16 (COM), PDFium. Bleibt genaehert (gemeldet): Kurven als Strecken (0,05 pt), Anker am Zeichen, wrapThrough wie wrapTight, hinter dem Text nur ohne Umfluss, Anfasser/extrusionOk nicht ausgewertet, perspektivische Kamera/Material/Reliefprofile; Excel mit Office-Vorgabethema, PowerPoint-Writer nur Balken/Linie/Torte und Vorlageformen teils als rect, Oberflaechendiagramme, ofPie/3D genaehert, Beschriftungen in Helvetica ohne manualLayout; im .kdoc fehlen Kopf-/Fusszeilen- und Fussnotentext; fette Schnitte mit duennem Strich, Times ohne Serifen im Rasterizer	

v4-LXL · Excel (.xlsx)

#	Excel-Funktion	Stand bei dir	Aufw.
LXL1	Formel-Auswertung (numerischer Rechenkern <code>XlsxFormula</code> : <code>SUM/AVERAGE/MIN/MAX/COUNT/IF/ROUND/ABS/VLOOKUP</code> + Arithmetik/Vergleiche/Bereiche) + Formeln schreiben (<code><f></code> , M1291); Text-Funktionen (<code>LEFT/RIGHT/MID/LEN/CONCATENATE/TRIM/UPPER/LOWER/TEXT</code>), Datums-Arithmetik (<code>DATE/YEAR/MONTH/DAY/EDATE/DATEDIF</code> , Excel-1900-System inkl. Schaltjahr-1900-Bug), benannte Bereiche in Formeln (<code>XlsxDefinedName</code> auflösbar) und die Neuberechnungs-Kette (<code>XlsxFormulaEngine.RecalculateAll</code> — Abhängigkeitsreihenfolge + Ringbezug-Erkennung) — ehrlich benannte Grenzen: kein <code>TODAY()/NOW()</code> (nicht deterministisch), <code>TEXT</code> nur einfache Nachkommastellen-Muster (keine Datumsformate)	M1291+	XL
LXL2	Zellformatierung lesen + rendern: <code>Fuellfarbe/Fett/Schriftfarbe</code> (M1291), Zellrahmen (<code>duenn/mittel/dick</code>) + Ausrichtung (<code>waagerecht/senkrecht</code>) nach Zellstil (M1291-Rest)		L
LXL3	Bedingte Formatierung (<code>cells/colorScale/dataBar/iconSet read/write</code> + im <code>xlsx->pdf</code> angewandt) + Datenprüfung (<code>list-Dropdowns</code> + <code>Zahl-/Datums-/Textlaengen-Bedingungen, read/write</code>) - komplett (M1293, M1358)	M1293/M1358	L
LXL4	Diagramme LESEN (Balken/Linie/Torte/Flaeche/Streu/Ring - Zeichnungs-Kette -> <code>chart-Teil</code>) + SCHREIBEN (<code>XlsxWriteSheet.AddChart</code> , eine Reihe je Diagramm) + als natives <code>ChartFrame</code> in <code>xlsx->pdf</code> gerendert (Flaeche/Ring als Näherung ueber Linie/Torte, ehrlich benannt)		L
LXL5	Benannte Bereiche (<code>definedName</code>) + Fensterfixierung (<code>pane</code>) + Autofilter (<code>autoFilter</code>) <code>read/write</code> (M1294) + Pivot-Tabellen lesen (Quellbereich, Zeilen-/Spalten-/Wertfelder) und im PDF als zur Laufzeit aus den Rohdaten gerechnete Kreuztabelle darstellen — nur <code>sum/count/average</code> als echte Aggregation, keine berechneten Felder, keine verschachtelten Mehrfach-Zeilenfelder (M1360)	M1294/M1360	M (Nische)
LXL6	Bilder im Blatt LESEN (Zeichnungs-Kette -> <code>xl/media</code>) + in <code>xlsx->pdf</code> gerendert (M1295) + SCHREIBEN (<code>XlsxWriteSheet.AddImage - xl/drawings + xl/media + rels</code> ; Diagramme und Bilder teilen sich EINE <code>drawingN.xml</code> je Blatt, ihre Beziehungen einen <code>rId-Raum</code>) - ehrlich benannte v1-Grenzen: nur PNG (die Bytes werden durchgereicht, nicht geprüft oder umgewandelt), nur der Anker-Typ <code>oneCellAnchor</code> (keine Größenanpassung an den Zellinhalt), keine Formen im Zeichnungsteil		M (Nische)

v4-LFO · Apache FOP (XSL-FO)

#	FOP-Funktion	Stand bei dir	Aufw.
LFO1	<code>fo:marker / retrieve-marker</code> (laufende Kolumnentitel aus dem Inhalt)	M1312 (<code>Fo/FoContext + HeaderFooterOptions.ResolvePerPage</code>): der Kolumnentitel wird je Seite aus dem Inhalt aufgelöst, ueber den vorhandenen <code>HeaderFooterStamper</code> . Grenzen: <code>retrieve-position</code> als <code>first/last</code> auf der Seite, keine <code>retrieve-boundary</code> -Feinheiten, Markerinhalt als Klartext	M
LFO2	Zell-Spans (<code>number-columns-spanned / number-rows-spanned</code>) + <code>background-color</code> je Zelle	M1281	M
LFO3	<code>fo:leader</code> M1285; <code>fo:float</code> + echte <code>fo:footnote</code> M1312 (ueber die vorhandene Umflussmechanik <code>FloatToTop/FloatToSide</code> M234/M1022 und <code>ParagraphWithFootnotes</code> M991 - nichts neu gebaut). Grenzen: <code>Float-Hoehe</code> per Messlauf, <code>before-Float</code> schiebt nur Folgeinhalt, Fussnote in einer <code>Change Bar</code> wird gemeldet	M1312	M

#	FOP-Funktion	Stand bei dir	Aufw
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/M1426 (Fo/FoOutput + MCP fo_to_png/fo_to_tiff/fo_to_ps/fo_to_pcl): PNG je Seite, mehrseitiges TIFF, PostScript Level 2, HP PCL5 monochrome Rastergrafik (Print-Stream fuer Direktdruck, rundlauffaehig ueber PclToPdf). Multi-page-sequence-Zusammenfuehrung, fo:declarations-Metadaten und PDF/UA-Tagging nach goldenem Standard. AFP als IBM-Grossrechner-Format bewusst nicht im Kern	L

v4-LTX · LaTeX (pdflatex/xelatex)

LTX3	Theorem-Umgebungen (\newtheorem) M1283; \substack und kommutative Diagramme (amscd \begin{CD} mit @>>>/@<<</@VVV/@AAA/@=/@//@. samt Pfeilbeschriftung) M1353 - beide ueber den vorhandenen Matrix-Baustein, kein zweiter Satzweg. Grenze: die Beschriftung steht als Hoch-/Tiefstellung am Pfeil, nicht exakt darueber zentriert	M1283/M1353	M	
LTX4	book/report-Klasse: \part/\chapter + Nummerierung (Kapitel.Abschnitt)	M1282	M	
LTX5	Beamer (Praentationsklasse)	M1355/M1359/M1420: \documentclass{beamer} setzt Folien - eine Seite je frame, Folienformat 128×96 mm bzw. 160×90 mm bei aspectratio=169, \frametitle und der Titel hinter \begin{frame}{...}. Overlays wirken wirklich: \pause rollt die Folie auf der Token-Ebene zu Zwischenfolien aus (M1359), dazu die Bereiche \item<2->, \only, \uncover, \visible, \onslide und \alert (M1420) - \alert blendet dabei nicht aus, es hebt hervor und steht auf jeder Zwischenfolie. \usetheme setzt eine Fussleiste, \setbeamertemplate bildet footline/headline/frametitle/navigation symbols ab. Gemeldet statt still: die Plus-Formen <+>/<.->, die Aktionsangabe <2 alert@2>, \setbeamercolor/-font, und \uncover blendet aus statt Platz frei zu lassen	M135	L
LTX6	fontspec/XeLaTeX + microtype	M1352: \setmainfont/\setsansfont/\setmonofont (auch BoldFont=/ItalicFont=) binden eine Systemschrift ein - opt-in ueber LatexConversionOptions.FontDirectories, sonst wird das Dateisystem nicht angefasst und der Bericht sagt es; eine nicht gefundene Familie wird benannt statt still ersetzt. microtype: der Randausgleich (Protrusion) wirkt, die Font-Expansion ist nicht umgesetzt (gemeldet)	M135	M

LTX3	Theorem-Umgebungen (<code>\newtheorem</code>) M1283; <code>\substack</code> und kommutative Diagramme (<code>\amscd \begin{CD}</code>) mit <code>@>>>/@<<</@VVV/@AAA/@=/@/</code> . samt Pfeilbeschriftung) M1353 - beide ueber den vorhandenen Matrix-Baustein, kein zweiter Satzweg. Grenze: die Beschriftung steht als Hoch-/Tiefstellung am Pfeil, nicht exakt darueber zentriert	M1283/M1353	M	
LTX1	TikZ/PGF	M1354/M1357/M1358/M1420/M1428: Koordinaten, Pfade, Knoten, <code>scope/foreach</code> , <code>positioning</code> , <code>Baeume</code> , Matrizen, <code>pgfplots</code> , Funktionsplots und logarithmische Achsen - M1428 dazu: eigene Achsengrenzen <code>xmin/xmax</code> und Teilstrichlisten <code>xtick</code> (auch auf der log-Achse, durch denselben <code>log10</code> -Weg), parametrische Plots (<code>{\cos(t)}</code> , <code>{\sin(t)}</code>) ueber denselben Ausdrucksrechner und damit dieselbe Winkeleinheit (Grad), Balken <code>ybar/xbar</code> , Fehlerbalken und Marker, bei den Baeumen <code>grow=</code> in Grad, <code>level distance=</code> / <code>sibling distance=</code> und der rechtwinklige <code>edge from parent fork down up left right</code> , und schliesslich ein eigener <code>edge from parent path={...}</code> : die Platzhalter <code>\tikzparentnode/\tikzchildnode</code> werden als benannte Knoten gebunden und der Ausdruck <code>laeuft</code> durch den vorhandenen Pfadleser - kein zweiter Auswerter; dabei fielen <code> </code> - <code>/</code> - <code> </code> und relative Bezier-Kontrollpunkte <code>+(dx,dy)</code> fuer jeden Pfad ab. Gemeldet statt geraten: eine Ankerangabe (<code>A.south</code>) trifft die Knotenmitte; ein Punkt ausserhalb der Achsengrenzen faellt ganz weg (die Linie bekommt eine Luecke, statt bis zum Rand gezogen zu werden); Flaechendiagramme (<code>fill between</code> , <code>stack plots</code>), gruppierte Balken, externe Datendateien, <code>grow'</code> und Abstaende je Ebene	M142	XL
LTX2	BibTeX/biblatex (.bib lesen + Zitierstile)	M1428: <code>Latex/BibTexParser</code> (alle vier Namensformen, <code>@string</code> mit #-Verkettung, verschachtelte Klammern) + <code>BibliographyFormatter</code> , Sortierung nach Autor/Jahr (<code>nyt/nty/none</code>) und die Suffixe <code>2020a/2020b</code> (M1420) - M1428 dazu: ein echter <code>.bst</code> -Interpreter (Stapelmaschine mit <code>ENTRY/FUNCTION/READ/ITERATE/SORT</code> , den <code>\$</code> -Operationen und der Formatsprache von <code>format.name\$</code>). Angebunden wie <code>\includegraphics</code> : die <code>.bst</code> wird nur gelesen, wenn ein Basisverzeichnis gesetzt ist und die Datei danebenliegt; ein abbrechender Stil faellt gemeldet auf die zwei eingebauten zurueck, ein <code>\bibliographystyle{plain}</code> ohne beiliegende <code>plain.bst</code> ist dagegen der Normalfall und wird bewusst nicht gemeldet. Fremdverifiziert: die vier Originalstile <code>plain/alpha/abbrv/unsrt</code> laufen unveraendert aus TeX Live durch und treffen die Ausgabe eines echten BibTeX 0.99d auf dieselbe <code>.bib</code> - im Testlauf laeuft weder TeX noch Docker, beides liegt als Fixture bei. Grenze: die Kollation ist genaehert statt sprachspezifisch nach ISO 14651	M142	L

v4-LHT · HTML/CSS (Browser-Rendering)

#	Browser-Funktion	Stand bei dir	Aufw.
LHT1	CSS Flexbox (<code>display:flex</code> , <code>flex-grow/wrap/order/align-self</code>)	M237/M643 (war faelschlich als Luecke gefuehrt)	L
LHT2	CSS Grid (<code>display:grid</code> , <code>auto-fill</code>)	M-Grid (war faelschlich als Luecke gefuehrt)	L
LHT5	<code>@media</code> (Typ + Breiten-Features <code>min/max-width</code>) M651; <code>position:sticky</code> als In-Fluss gerendert (Scroll ist kein PDF-Konzept)	M651 (war faelschlich als Luecke gefuehrt)	M

#	Browser-Funktion	Stand bei dir	Aufw.
LHT6	CSS Custom Properties (--x/var()) + calc()/min/max/clamp	M887/M888 (war faelschlich als Luecke gefuehrt)	S

v4-LPP · PowerPoint (.pptx)

#	PowerPoint-Funktion	Stand bei dir	Aufw.
LPP2	Tabellen im Foliensatz (graphicFrame/tbl) lesen + schreiben		M
LPP5	Notizen (notesSlide) + Folien-Kommentare lesen/schreiben	(Notizen ohne Folienbild-Platzhalter, klassisches p:cm-Kommentarmodell, beides im pptx->pptx-Rundlauf erhalten)	S
LPP6	Themen (theme1.xml: Farb- und Schriftschema) statt Standard-14-Abbildung	(Farbschema aufgeloeset - schemeClr in Laeufen, explizite srgbClr hat weiter Vorrang; Schriftschema gelesen, noch nicht auf den Textsatz angewandt)	M
LPP7	Bild-Zuschnitt (blipFill/srcRect), Gruppen (grpSp), Drehung/Spiegelung	(Zuschnitt/Drehung/Spiegelung gelesen + im PDF gezeichnet, Bildpunkt-verifiziert; Gruppen beim Lesen auf absolute Folienkoordinaten aufgeloeset - der Rundlauf schreibt sie flach zurueck)	M
LPP9	Uebergaenge/Animationen - nur fuer den pptx->pptx-Rundlauf sinnvoll (im PDF nicht darstellbar)	(PptxPresentationWriter: p:transition/p:timing roh erhalten; PptxToPdf meldet sie ehrlich statt sie stillschweigend zu uebergehen)	M (Nische)

v4-LVS · Visio — in M1436 geschlossene Scheiben

#	Punkt	Stand	Aufw.
LVS8	Darstellung, Rest: eingebettete Bilder (Raster ueber RasterImage.Parse, EMF/WMF als Vektoren ueber Metafile) auf das Shape beschnitten, im .kdoc als ImageFrame bzw. PdfPageFrame, beim Export als Fremdobjekt unter visio/media, auch aus .vdx; Fuellmuster 2..40 (Schraffuren und alte Verlaeuft), Verlaeuft nach Visio 2013, Transparenz von Flaechen und Linie, Schatten — in PDF, .kdoc (fillHatch, Verlauf, Deckkraft, Schlageschatten) und zurueck	M1436, gegen LibreOffice (wo es sie auswertet; sonst Office-Konvention, benannt)	L
LVS1	Themen und Ebenen: mehrere Themen-Teile, Farbschemata und Varianten je Seite, Farbabwandlungen ueber DrawingMIColors, Stil-Matrix samt Themen-Strichstaerke, Linienfarbe ueber die ganze Stilketten, Verbinder-Format des Themas; Ebenen im Objektmodell (LayoutObject.LayerNames, PdfLayer mit „Gesperrt“), im .kdoc (layers), im Visio-Rundlauf und als Editor-Palette (sichtbar/druckbar/gesperrt)	M1436	M
LVS1	.vsd der Fassungen 5 und 6 (Visio 5 bis 2002) — dazu zwei Fehler im Fassung-11-Weg behoben: die Linienzeile wurde ein Byte versetzt gelesen (jede Linie unsichtbar), und der Zeichnungsmaassstab fehlte	M1436, Pruefdateien aus dem libvisio-Bestand (MPL-2.0)	M

#	Punkt	Stand	Aufw
LVS1	Zeichnungsmaassstab im .vsdx: PageScale/DrawingScale rechnen Lage, Groesse, Geometrie (samt absoluten NURBS/POLYLINE-Punkten), Textblock, Bildlage und Seitengroesse in Papiereinheiten um — eine Regel VisioMaassstab fuer .vsd und .vsdx; dabei behoben, dass eine Seite ueber 500 Zeichnungszoll still auf 8,5 x 11 zurueckfiel	M1436, gegen LibreOffice	S

v4-LVS · Visio — in M1435 geschlossene Scheiben

#	Punkt	Stand	Aufw
LVS7	Freie Kurven exakt: NURBSTo (polynomial exakt ueber die Hermite-Form je Knotenspanne; rational oder hoehergradig adaptiv mit gemessener Abweichung und VSDX_GEOMETRIE_GENAEHERT), PolylineTo, SplineStart/SplineKnot, InfiniteLine an der Seite beschnitten	M1435, gegen LibreOffice	M
LVS9	Text, Rest: fette Standard-14-Schnitte im eigenen Rasterizer wirklich fett (der Leser war richtig, mager war das Bild); Absatzformat je Absatz, leere Absaetze und Visio-Grundlinien im .kdoc; die genannte Schrift opt-in eingebettet ueber VsdConversionOptions.FontDirectories (wie fontspec im LaTeX-Weg), ohne Ordner Standard-14 mit VSDX_SCHRIFT_ERSETZT	M1435	M
LVS1	Ebenen: Layer (Name/Visible/Print) und LayerMember; ein Shape auf nur unsichtbaren Ebenen wird nicht gezeichnet, auf nur nicht druckbaren nicht ins PDF (die Druckausgabe); gemeldet	M1435 (LibreOffice wertet das Druck-Kennzeichen nicht aus, es gilt Visios Zellbeschreibung)	M
LVS1	Visio-Gruppen erhalten: im Lesemodell VsdPage.Groups, im .kdoc verschachtelte ObjectGroups mit dem aus den Kind-Formeln gelesenen Groessenverhalten, beim Export Type='Group' mit den Formeln, die Visio selbst schreibt; nicht eindeutige Formeln bleiben proportional und werden gemeldet (VSDX_GRUPPENVERHALTEN_UNKLAR)	M1435, LibreOffice zeichnet den Rueckweg an denselben Stellen	M
LVS1	Export vollstaendig: ein Verbinder geht als 1-D-Verbinder mit _WALKGLUE/<Connect> an die benannten Formen; ein zusammengesetzter Pfad mehrere Geometry-Abschnitte, gerade-ungerade gefuellt (eine Regel VsdGeometrien.Verbund fuer PDF und .kdoc)	M1435	M
LVS1	Kurven im .kdoc: neu ShapeKind.Path mit ShapePathSegment (M/L/C/Z) und Even-Odd, im .kdoc als path/evenOdd, gezeichnet mit echten Kurven, Export als CubBezTo	M1435	M
LVS1	Alte Formate: .vdx (XML, in die .vsdx-Zellform uebersetzt) und .vsd Fassung 11 (OLE-Verbunddokument, LZ77, Chunks) — beide laufen durch den vorhandenen VsdReader; in DocumentConverter und den MCP-Werkzeugen erreichbar	M1435, Pruefdateien aus dem libvisio-Bestand (MPL-2.0), Sollwerte aus LibreOffice	L
LVS1	Feinabgleich: graue Titelbalken sind richtig (Visios eigenes Vorschaubild zeigt Grau, LibreOffice Weiss — es gilt Visio); Themen-Linienfarbe ueber QuickStyleLineColor; Linienenden wachsen linear mit der Strichstaerke (an LibreOffice gemessen); Themed-Zeichenfarbe wird dk1	M1435	S

v4-LVS · Visio — in M1434 geschlossene Scheiben

#	Punkt	Stand	Aufw.
LVS8a	Strichmuster und Linienenden: LinePattern 2..23 in Vielfachen der Strichstaerke, die 45 Visio-Linienenden (Pfeile, Kraehenfuesse) in beide Richtungen — Modell, PDF, .kdoc (startLineEnd/endLineEnd), Export	M1434, Sollwerte aus LibreOffice	M
LVS9a	Text originalgetreu: cp/pp/tp/fld, Zeichen- und Absatzformate zellweise geerbt, Textblock mit Raendern, senkrechter Ausrichtung, Hintergrund und Drehung, Umbruch ueber TextLayout; im .kdoc ein formatierter Rahmen je Textblock, Rueckweg mit cp/pp	M1434, Grundlinien gegen LibreOffice im Median unter 0,25 pt	M
LVS10	Geometrie-Vererbung, Stilketten, Themenfarben: Master zellweise verschmolzen (Del nimmt Geerbtes weg, Gruppen-Glieder ueber MasterShape), immer der Wert V, Stilvorlagen-Kette mit Themed, Themenfarben ueber QuickStyle, Verbinders mit <Connects>, Ecken-Rundung	M1434	M
LVS16	Gruppen bearbeiten und Groessenverhalten je Kind: proportional, fest am Anfang/Ende/mittig oder strecken, je Achse; Editor (in die Gruppe gehen, Kind ziehen/aendern/aufnehmen/herausnehmen, Gruppe ueber die Anfasser skalieren), .kdoc (groupResize), Generator/REST	M1434	M
LVS17	Ausgabe aus .kdoc: TIFF (alle Seiten), PNG/JPEG/BMP/GIF/WebP (Seite 1), EMF/WMF (vektoriell), wahlbare Aufloesung; .vsdx und .idml direkt als Arbeitsdokument statt ueber ein Zwischen-PDF	M1434	M

v4-LVS · Visio (.vsdx) — in M1429 geschlossen (6 von 6)

Neu gebaut, es gab vorher keine Zeile Visio-Code. Modul Visio, Zero-NuGet (System.IO.Compression fuers OPC/ZIP-Paket, System.Xml.Linq fuers XML, XXE-fest). Gegen zwei unabhaengige Werkzeuge belegt — LibreOffice Draw (libvisio) und die Python-Bibliothek vsdx; im Testlauf laeuft keines von beiden, die Fixtures liegen eingebettet bei.

#	Visio-Funktion	Stand bei dir	Aufw	
LVS1	VSDX-Leser (VsdxReader)	M1429: OPC/ZIP-Paket, Seiten, Shapes, Geometrien (MoveTo/LineTo/ArcTo/Ellipse/EllipticalArcTo/CubBezTo/QuadBezTo samt den Rel...-Formen, die Bruchteile statt Zoll tragen), Text, Stilvorlagen und die Vererbung vom Master. Boegen und Ellipsen werden zu kubischen Bezierkurven aufgeloeset, damit Drehung (Angle) und Spiegelung (FlipX/FlipY) auf alles gleich wirken; die Lage kommt aus Pin LocPin. Gegen LibreOffice nachgemessen auf ein Zehntel Punkt. Wirft bei keinem Eingabebyte. Gemeldet statt still: NURBSto/PolylineTo/SplineKnot/InfiniteLine werden zur geraden Strecke genaehert (VSDX_GEOMETRIE_GENAEHERT); Bilder, Fuellmuster, Verlaeuft, Schatten, Strichmuster, Pfeilspitzen, Themen und Ebenen werden nicht ausgewertet; Text einzeilig in Helvetica ohne Umbruch und Drehung	M142	L

#	Visio-Funktion	Stand bei dir	Aufw	
LVS2	VSDX nach PDF (VsdnToPdf) + MCP vsdn_to_pdf	M1429: gemalt ueber die vorhandene PageCanvas-Vektor-Kette - kein zweiter Zeichenweg, wie Metafile und Svg. Die Umrechnung Zoll->Punkt geschieht an genau einer Stelle. Es gelten die Leser-Grenzen; ein Zeichen, das die Standardschriften nicht fuehren, wird weggelassen und gemeldet (VSDX_TEXT_NICHT_DARSTELLBAR) - ein Fragezeichen saehe nach Absicht aus	M142	M
LVS3	VSDX-Schreiber (VsdnWriter) + PDF nach VSDX (PdfToVsdn) + MCP pdf_to_vsdn	M1429: das OPC-Paket mit [Content_Types].xml, den _rels, document.xml, pages.xml und je Seite ein pageN.xml. PdfToVsdn liest die Seite ueber die Anzeigeliste und den Textextraktor - wie PdfToMetafile und aus demselben Grund: PageGraphicsExtractor liefert nur umschliessende Rechtecke und gar keine Farben. Die Achse wird nicht gedreht (Visio und PDF zaehlen beide von links unten nach oben). Gemeldet statt still: Bilder, Verlaeufe, Transparenz, Beschnitt	M142	L
LVS4	Direktimport: .vsdx als editierbares .kdoc (VsdnToKdoc) + MCP vsdn_to_kdoc	M1429: direkt, nicht ueber ein zwischengerendertes PDF - dort waere aus einem Rechteck ein Pfad aus vier Strecken geworden und aus dem Text ein Haufen Laeufe, also nichts mehr Anfassbares. Rechteck -> ShapeKind.Rectangle, Ellipse -> Ellipse, Zweipunktstrecke -> Line, sonst Polygon, mehrere geschlossene Teilpfade -> CompoundPath. .vsdx ist in DocumentConverter angeboten UND wirklich wandelbar (die Fehlerklasse IMG1/IMG22/IMG27). Dabei gefunden: Visio legt ein Rechteck ohne Close ab (der letzte LineTo geht zum Anfang zurueck) - wer nur auf den Close sieht, bekommt ein Polygon mit fuenf Punkten. Gemeldet statt still: freie Kurven werden geflacht (VSDX_KDOC_KURVE_GEFLACHT), Text einzeilig ohne Schnitt und Farbe, Gruppen loest schon der Leser auf	M142	M
LVS5	Konnektoren mit Klebepunkten im Editor	M1429: ein Konnektor speichert seine Endpunkte nicht, sondern haelt den Namen des Start- und Ziel-Shapes samt Klebepunkt und rechnet die Stelle beim Zeichnen - nur so haelt er beim Verschieben. Dasselbe Muster, mit dem MasterPage den Namen des Masters haelt statt seiner Elemente. Klebepunkte an den Seitenmitten, Ecken und der Mitte, automatische Seitenwahl nach der Lage des Ziels, gerader und rechtwinkliger Verlauf, im .kdoc-Rundlauf gesichert	M142	L
LVS6	Editor-Export nach .vsdx (KdocToVsdn) + MCP kdoc_to_vsdn	M1429: ebenfalls direkt aus dem Objektmodell, nicht ueber ein Zwischen-PDF. Im DocumentExporter eingehaengt, damit Editor-Knopf und kdoc_to_document dieselbe Stelle bedienen. Zwei Fehler dabei gefunden, einer nur durch das Fremdorakel: der Verbinder sass um seinen eigenen Ursprung daneben (ResolvedPoints liefert bereits absolut, X/Y wurden ein zweites Mal addiert) - die eigenen Tests prueften Art und Anmeldung, nicht die Stelle, gefunden hat es die vsdx-Bibliothek an der wirklich geschriebenen Datei; und der VsdnReader suchte die Schriftgroesse nur am Shape und in der Stilvorlage, waehrend eine echte Visio-Datei sie in <Section N='Character'> ablegt - jede Datei lieferte damit nur den Vorgabewert. Gemeldet statt still: Bilder, Tabellen, die uebrigen Fliess-Rahmen; ein Verbinder wird ein freier Streckenzug ohne Klebepunkte und Pfeilspitzen; ein zusammengesetzter Pfad verliert die Even-Odd-Fuellung; Drehung, Deckkraft, Effekte, Verlaeufe	M142	M

v4-IMG · Bildformate — Lesen und Schreiben

Format	Lesen	Schreiben	Bemerkung
JPEG	eigener Codec		inkl. progressiv
PNG			indiziert + TrueColor
TIFF	voll	voll	Lesen: unkompr./LZW/Deflate/PackBits/CCITT G3+G4/JPEG-in-TIFF, 1–16 Bit, BigTIFF (M1332) · Schreiben: RGB/RGBA/Grau, unkompr./Deflate/PackBits/LZW/CCITT G4, mehrseitig (M1298/M1331)
GIF			LZW/Interlace/Transparenz, Median-Cut, auch animiert (M1300/M1335)
HEIF/HEIC	echte Pixel (M1147), Kachelgitter + Deckkraft + Tiefenkarte (M1339/M1340)		eigener HEVC-Intra-Decoder; schreiben -> IMG7
JPEG 2000		verlustfrei (M1306) + verlustbehaftet (M1344, ohne Ratensteuerung)	eine echte Zielgrösse braucht den PCRD-Optimierer -> IMG17
WebP	verlustfrei und verlustbehaftet (M1003)	verlustfrei (M1301) und verlustbehaftet (M1400)	animiert in beide Richtungen (M1337); die Deckkraft nur verlustfrei
JBIG2		generische Region (M1345)	Einbettung ins PDF ueber recompress_bilevel (M1346); Symbol-Woerterbuch -> IMG23
CCITT G3/G4		G4 (M1331)	Filter-Ebene; G3 wird gelesen, aber nicht geschrieben
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 Grössen, Klickpunkt; .ani nur lesen
ICNS (macOS)	(M1400)	(M1400)	RLE kanalweise, ARGB, eingebettetes PNG/JPEG 2000
Symbole aus .exe/.dll	(M1400)	–	PE-Ressourcenbaum -> .ico
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
RAW/DNG	Sensordaten + Entwicklung (M1309)	x	inkl. verlustfreiem JPEG (SOF3); schreiben ist nicht vorgesehen
JPEG XL	Behälter, Bildkopf, Rahmenkopf und modularer Pfad - liefert Bildpunkte (M1408)	x	Schreiben -> IMG8.6; der VarDCT-Pfad (verlustbehaftet) -> IMG8.5

#	Bildformat-Funktion	Stand bei dir	Aufw.
IMG1	BMP lesen — der gemeldete Fehler ist behoben: IRasterImage.Parse erkennt die BMP-Signatur und dekodiert ueber Imaging/BmpDecoder (1/4/8/16/24/32 Bit, Palette, BI_BITFIELDS, bottom-up und top-down); Lauflaengen-Kompression BI_RLE8/BI_RLE4 in beide Richtungen und Palettenbilder schreiben (M1334)	M1297/M1308/M1334	S
IMG2	TIFF schreiben in Farbe + mehrseitig: Imaging/TiffWriter (RGB/RGBA/Graustufen, unkomprimiert, Deflate, PackBits, LZW und CCITT G4 aus M1331, mehrere Seiten ueber verkettete IFDs) — ehrlich benannte Grenzen: 8 Bit je Kanal (16 Bit und CMYK offen), ein Streifen je Seite, keine Vorhersage-Filter	M1298/M1331	M
IMG3	GIF schreiben: Imaging/GifEncoder (GIF89a, globale Farbtabelle, GIF-LZW, Transparenzindex, Median-Cut ab 256 Farben) — auch animiert (M1335: mehrere Einzelbilder, Standzeiten, Wiederholungszahl)	M1300/M1335	M
IMG4	WebP schreiben: EncodeLossless (VP8L, bit-genau inkl. Teiltransparenz), animiert (M1337) und seit M1400 auch verlustbehaftet (EncodeLossy — Einzelheiten unter IMG16)	M1301/M1337/M1400	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
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)	M1304 (PsdDecoder): kanalweise, unkompr./PackBits/ZIP, Graustufen/RGB(+Alpha)/CMYK (invertiert)/indiziert, 1/8/16 Bit; von Pillow gegengelesen. Grenzen: keine Ebenen/Masken/Effekte, kein 32-Bit-Gleitkoma	M (Nische)
IMG1	ICO / TGA / PNM (PBM,PGM,PPM) lesen	M1302 (IcoDecoder/TgaDecoder/PnmDecoder): ICO/CUR mit AND-Maske, TGA unkompr. und laulaengen-kodiert, Netpbm P1-P6; von Pillow gegengelesen. Dabei die Erkennungs-Falle geloest, dass ein unkomprimiertes TGA wie ein Zeiger aussieht	S (Nische)
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 Entmosaikern. 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)	M1303 (ImageConversion/ImageConverter + MCP convert_image): gelesen wird alles, was RasterImage.Parse kennt, geschrieben PNG/JPEG/TIFF (auch mehrseitig)/GIF/WebP/BMP/TGA/PNM/ICO/PSD/JPEG 2000; optionales Verkleinern ueber MaxEdge. Die Wandlung liegt in einem eigenen Kompositionsmodul, weil Imaging das Modul Content nicht importieren darf	S

#	Bildformat-Funktion	Stand bei dir	Aufw.
IMG1	EXIF lesen und schreiben	M1400 (Imaging/ExifCodec auf dem vorhandenen TiffDirectory, kein zweiter IFD-Leser; ImageMetadataCodec als Fassade): Aufnahmezeit, Kamera/Objektiv, Belichtungszeit, Blende, ISO, Brennweite, Ausrichtung und GPS in JPEG-APP1, PNG-eXIf und im IFD0 einer TIFF. Fixtures aus Pillow. Ueber MCP image_metadata/set_image_metadata. Grenzen: TIFF nur lesend (die Bilddaten unangetastet zu lassen waere ein eigener Schritt); nur eine praxisnahe Tag-Teilmenge; das Anwenden der Ausrichtung (Drehen) ist nicht Teil des Codecs	M
IMG1	IPTC/XMP im Bild lesen und schreiben	M1400 (Imaging/IptcCodec fuer die IIM-Datensaetze in der Photoshop-8BIM-Ressource, XmpImagePacket als XXE-fester RDF-Leser/-Bauer): Titel, Beschreibung, Urheber, Rechte, Stichwoerter in JPEG (APP1/APP13), PNG (iTXt), TIFF (Tag 700, lesend) und WebP (XMP-Block inkl. VP8X-Erzeugung). Geschrieben wird IPTC UND XMP aus derselben Quelle - aeltere Werkzeuge lesen nur das eine, neuere nur das andere. Grenzen: ein JPEG mit vorhandener Photoshop-Ressourcensammlung verliert deren uebrige Eintraege; komprimiertes iTXt wird uebergangen	M
IMG1	WebP verlustbehaftet schreiben (VP8-Intra-Encoder)	M1400 (WebPEncoder.EncodeLossy + Imaging/Vp8Encoder): boolescher Entropie-Codierer mit Uebertrags-Fortpflanzung, Rahmenkopf, Intra-Vorhersage (16x16 DC/V/H/TM, alle zehn 4x4-Modi, 8x8-Chroma), 4x4-DCT + Walsh-Hadamard, Quantisierung, Token-Kodierung - vorhergesagt wird gegen die rekonstruierten Nachbarn. Geprueft ueber den eigenen Vp8LossyDecoder und libwebp (Guete 90: 1,05 mittlere Abweichung je Kanal). Grenzen: Schleifenfilter aus, keine Segmentierung, eine Token-Partition, keine Wahrscheinlichkeits-Anpassung, keine Ratensteuerung, keine Deckkraft (kein ALPH-Block)	XL
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 aufgelosten Farben, nicht die blossen Rechtecke des Grafik-Extraktors). Dabei zwei stille Leser-Fehler behoben: Pinsel und Stift faerbt 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	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-Blocke (bfAttributes Bit 1) werden uebersprungen statt geraten	S–M
IMG2	macOS-Symbole (.icns) lesen und schreiben	M1400 (IcnsDecoder/IcnsEncoder): is32/il32/ih32/it32 (RGB, roh pixel-verschachtelt oder Apple-RLE kanalweise - ein am fremden Leser gemessener Unterschied, den ein diskriminierender Test faengt) samt Maskenblocken, ic04/ic05 (ARGB) und ic07-ic14 (eingebettetes PNG/JPEG 2000 ueber die vorhandenen Decoder). In RasterImage.Parse, ImageConverter und convert_image erreichbar. Grenzen: nur die sieben Standardgroessen, nur quadratisch	M
IMG2	Symbole aus Programmen holen (.exe/.dll)	M1400 (Imaging/PelconExtractor + MCP extract_exe_icon): PE32/PE32+-Ressourcenbaum, RT_GROUP_ICON/RT_ICON ueber den vorhandenen IcoEncoder zu einer .ico zusammengesetzt - samt der zwei Fallen (der Gruppeneintrag traegt eine Ressourcen-ID statt eines Versatzes, und die Versaetze sind RVAs). Real geprueft an notepad.exe und explorer.exe. Grenze: auf neueren Windows-Fassungen liegen die Ressourcen mancher Systemdateien ausgelagert (.mun) und fehlen im Hauptmodul - bei shell32.dll gemessen	M

#	Was fehlt	Warum es vorkommt	Aufw
IMG27	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 Praediktor, 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
IMG28	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
IMG30	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
IMG31	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
IMG32 11/11	JPEG 2000: das volle Codestream-Profil (Kacheln, Schichten, Precincts, Behaelter-Boxen)	Untertabelle unten: IMG32.1–32.11 — seit M1408 vollstaendig	L
IMG33	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

Die grossen Bildformat-Punkte im Einzelnen

#	Scheibe	Stand
IMG6.1	OBU-Schicht, Sequenzkopf, av1C-Box	M1310 - von libavif gegengelesen
IMG6.2	Rahmenkopf (Einzelbild-Pfad)	M1400
IMG6.3	Arithmetischer Symbol-Decoder + Vorgabe-CDF-Tabellen	M1400/M1401
IMG6.4	Kachelgruppe und Partitionsbaum	M1400
IMG6.5	Intra-Vorhersage (alle 13 Modi) + Dequantisierung	M1400
IMG6.6	Inverse Transformationen (DCT/ADST/Identitaet/WHT, 2D)	M1402 - gegen die geschlossene Kosinus-/Sinus-Form geprueft, 0,006 % Abweichung

#	Scheibe	Stand
IMG6.7	Koeffizienten-Auslesung (§5.11.39) und Zusammenbau je Block	M1408 - Abtastreihenfolgen, Vorgabe-Wahrscheinlichkeiten und der Koeffizienten-Leser; die Bildpunkte haengen jetzt allein an IMG6.8

#	Scheibe	Stand
IMG7.1	HEIF-Behaelter (ftyp/meta/iinf/iloc/iprp/ipco/ipma/mdat)	M1400
IMG7.2	CABAC-Codierer	M1400
IMG7.3	Vorwaerts-Transformation und Quantisierung	M1400
IMG7.4	Parametersaetze VPS/SPS/PPS + Slice-Kopf, Intra-Encoder	M1400 - heif-convert (libheif) liest den Strom
IMG7.5	Deckkraft als Hilfsbild (auxl)	M1400
IMG7.6	Sammlungen und Kachelgitter (grid) schreiben	M1400

#	Scheibe	Stand
IMG8.	Behaelter (Codestream FF 0A und ISO-BMFF), Bitleser, Bildkopf	M1400 - Feldreihenfolge aus der libjxl-Quelle 0.12.0, 17 Pruefdateien von libjxl erzeugt
IMG8.	Rahmenkopf und TOC	M1408 - dabei fehlten im Bildkopf zwei Felder, ohne die der Rahmenkopf dahinter nicht an seiner Stelle begann; der Bitleser wurde vom ref struct zur Klasse, damit Kopf- und Entropieschicht denselben Leser teilen statt zweier Fassungen derselben Bitarithmetik
IMG8.	Entropie-Decoder (Praefix-Codes + rANS, Hybrid-Ganzzahlen, Kontext-Abbildungen)	M1408 - wie angekuendigt zusammen mit IMG8.4 belegt: libjxl exportiert keine Zwischenzustaende, also ist der Beleg das fertige Bild
IMG8.	Modularer Pfad (MA-Baeume, Selbstkorrektur-Praediktion, Squeeze/RCT)	M1408 - damit liefert JPEG XL Bildpunkte: .jxl steht jetzt in der Tabelle des Erreichbarkeits-Waechters statt in dessen Ausnahmeliste, kommt also ueber RasterImage.Parse UND DocumentConverter an

#	Scheibe	Stand
IMG23.1	Generische Region, arithmetisch (Seiteninfo, Region, Seitenende)	M1345 - kleiner als der eigene CCITT-G4-Schreiber
IMG23.2	Einbettung ins PDF (PdfBilevelRecompressor + recompress_bilevel)	M1346 - dabei zwei Fehler gefunden: das /Decode gehoert NICHT dazu, und der eigene Leseweg uebergang es
IMG23.3	Symbol-Woerterbuch + Textregion + Restbild-Region	M1400 - Textseite auf ein Drittel (190 statt 511 Bytes)
IMG23.4	Fremdverifikation und Freigabe	M1402 - zwei Ursachen gefunden (SBSYMCODELEN ist bei EINER Form 0, nicht max(1,...); die Endmarke FF AC fehlte); gegen MuPDF/big2dec und PDFium je 0 abweichende Bildpunkte, Vorgabe wieder AN
IMG23.5	MMR-Kodierung als Alternative zur arithmetischen	M1408 - ueber den vorhandenen Filters/CcittFaxEncoder, kein zweiter Codierer
IMG23.6	JBIG2Globals — ein Woerterbuch fuer mehrere Seiten	M1408 - der eigentliche Gewinn bei einem mehrseitigen Scan: das Symbolwoerterbuch steht einmal im Globals-Strom statt in jeder Seite

#	Scheibe	Stand
IMG29.1	12 Bit je Komponente lesen (SOF1)	M1338 - Fixture von libjpeg-turbo; Ausgabe auf 8 Bit heruntergerechnet (benannte Grenze)
IMG29.2	Progressiv schreiben (SOF2, Spektralauswahl)	M1343 - libjpeg liest es als progressiv und pixelgleich
IMG29.3	Arithmetische Kodierung lesen, sequenziell (SOF9)	M1400 - erst das Fremdorakel zeigte drei Fehler (Abweichung 224-255 je Kanal, obwohl alle eigenen Tests gruen waren)
IMG29.4	SOF10 — progressiv arithmetisch	M1408
IMG29.5	Restart-Intervalle in arithmetischen Scans	M1408

#	Scheibe	Stand
IMG32.	JP2-Behaelter schreiben (Signaturbox/ftyp/jp2h/jp2c)	M1341 - dabei ein stiller Fehler behoben: .jp2 bekam den rohen Codestream
IMG32.	Kacheln lesen und schreiben	M1342 - je Kachel eigene Wavelet und Teilbaender, Randkacheln kleiner
IMG32.	Ratensteuerung (PCRD-opt, echte Zielgroesse in Bytes)	M1400
IMG32.	Mehrere Qualitaetsschichten	M1400
IMG32.	Mehrere Teilstuecke je Kachel (Tile-Parts)	M1400
IMG32.	Behaelter-Boxen res /colr+ICC/cdef samt Deckkraft	M1400
IMG32.	Kleine Precincts lesen und schreiben	M1400 - dabei ein stiller Lesefehler geschlossen: eine fremde Datei mit Precincts kam falsch heraus (Bildpunkt 0 statt 17), ohne dass ein einziger eigener Test rot wurde
IMG32.	Mehrkomponenten-Transformation (RCT reversibel, ICT irreversibel)	M1405
IMG32.	Andere Progressionsverlaeuftypen als LRCP (RLCP, RPCL, PCRL, CPRL)	M1406 - alle fuenf, lesen und schreiben
IMG32.	Bittiefen ausser 8 (1–16 Bit, mit und ohne Vorzeichen)	M1407 - 12 und 16 Bit las der Dekoder laengst richtig (die Zeile war zu pessimistisch, gegen zwei opj_compress-Dateien belegt). Der Fund lag bei 38 Bit, die der Standard zulaesst: 1 << 37 maskiert C# stillschweigend auf 1 << 5 - die Datei wurde weder abgelehnt noch richtig gelesen. Ueber 31 Bit wird jetzt benannt abgelehnt. Schreibseite 1..8 (mehr hiesse, Abstufungen zu erfinden, die Rgbalmage nicht traegt)
IMG32.	Ein Precinct-Mass je Aufloesungsstufe beim Schreiben; Precinct-Raster am Bild- statt am Kachelursprung	M1408

v4-FNT · Schriften — alle Arten, beide Richtungen

Art	Lesen/Rendern	Einbetten/Subsetten	Als Datei ausgeben
TrueType (glyph)			.ttf byte-treu

Art	Lesen/Rendern	Einbetten/Subsetten	Als Datei ausgeben
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)	M1323	nur sbix-PNG	lesen+zeichnen, schreiben offen
Farbfonts SVG-in-OpenType	M1326	(ColorFontSvgEncoder, FED13-Rest)	lesen+zeichnen (als Vektor, nicht gerastert)
TrueType-Hinting (fpgm/prep-Bytecode)	(opt-in im Rasterizer)	—	—
Multiple Master · BDF/PCF/.fon/.fnt	M1316/M1347	BDF+PSF (PCF/FON offen)	

#	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
FNT2	Bitmap-Farbfonts 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

#	Schrift-Funktion	Stand bei dir	Aufw.
FNT4	TrueType-Hinting (Bytecode-Interpreter fpgm/prep/glyf-Instruktionen) fuer scharfe kleine Groessen im Rasterizer	Fonts/TrueTypeHinting: nahezu der volle Befehlssatz (Stack/Arithmetik/Logik/Kontrollfluss IF/ELSE/EIF/JMPR/JROT/JROF, beide Zonen - Glyph und Twilight -, alle Rundungszustaende inkl. SROUND/S45ROUND, MDAP/MIAP/MDRP/MIRP/SHP/SHC/SHZ/SHPIX/ALIGNRP/ISECT/IUP, DELTAP1-3/DELTAC1-3, WS/RS, FDEF/LOOPCALL/CALL). Eine wirklich nicht umgesetzte Instruktion (SDPVTI/IP/MSIRP/ALIGNPTS/FLIPPT/WCVTF/IDEF/GPV/GFV - die verbleibende, benannte v2-Luecke) bricht weiter benannt ab, statt zu raten. Angebunden an GlyphRasterizer.Rasterize (Parameter hinting, Vorgabe aus - ohne ihn byte-identisch); wirkt auf die bereits geglaetteten Umriss-Punkte (ehrlich benannte Naeherung, kein Zugriff auf die rohen glyf-Kontrollpunkte), schlaegt ein Programm fehl, faellt der Rasterizer auf den ungehinteten Umriss zurueck statt zu werfen	L
FNT5	Werkzeug convert_font - Schriftdatei in alle Richtungen wandeln (ttfotfwoffwoff2pfb/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	M1316 (Fonts/BitmapFont: BDF, PCF, Windows .fnt/.fon - FNT-Fassung 2.0/3.0 + NE-Behaelter, mehrere Groessen je Datei; gegen 192 reale .fon-Dateien gehalten). Diese Zeile stand faelschlich auf „fehlt“ - nachgezogen (M1316 hatte sie gebaut, aber nicht hier vermerkt)	S (Nische)

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

Format	Wofuer	Lesen	Schreiben
TTF (sfnt/glyf)	Web, Druck, System		(SfntWriter, FontBuilder - FED2)
OTF (sfnt/CFF)	Web, Druck		(CffOpenTypeWrapper, FontBuilder - FED2)
WOFF	Web		M1311 (WoffEncoder)
WOFF2	Web		M1311 (Woff2Encoder)
Variable Fonts (fvar/gvar/HVAR)	Web, Druck	inkl. Instanzieren	FED9 (VariableFontBuilder - fvar/gvar/avar/STAT; kein HVAR, die Breiten variieren ueber die Phantompunkte in gvar)
CFF2	Druck		(Cff2Writer - VarStore/blend aus statischen CFF-Mastern, die CFF-Entsprechung zu FED9s glyf/gvar)
CID-keyed CFF (ROS)	CJK-Druck		FED10 (CidKeyedCffWriter - Identitaets-Charset CID=GID, ein FArray-Eintrag)

Format	Wofuer	Lesen	Schreiben
Type1 (PFB/PFA)	Alter Druck		(Type1FontConverter)
Type3	PDF-intern		(PDF-Erzeugung)
Type42	PostScript		FED11 (Type42Writer, strukturell geprueft - kein Ghostscript auf dieser Maschine)
TTC/OTC (Sammlungen)	System	FED12 (FontCollection.Read)	FED12 (FontCollection.Write, Tabellen-Dedup)
COLR/CPAL (Farbe)	Web, System	v0+v1	v0 (ColorFontEncoder, M1364) + v1 (ColorFontV1Encoder, FED13-Rest)
CBDT/EBDT · sbix (Bitmap-Emoji)	System	(M1323)	(ColorBitmapFontEncoder - CBDT-Bildformat 17 bzw. sbix-Tag "png ", je genau eine Aufloesungsstufe; Bildformate 18/19, monochrome Formate und "dupe"/"jpg"/"tiff"/"pdf " bei sbix bleiben aussen vor)
SVG-in-OpenType	Web	(M1326)	(ColorFontSvgEncoder, FED13-Rest)
EOT	Web (alt, IE)		
SVG-Fonts	Web (abgeloeset)		
BDF · PCF · PSF	Linux/Konsole	(M1316 BDF/PCF, M1364 PSF1/PSF2)	FED15 (BitmapFontWriter.WriteBdf/WritePsf1/WritePsf2/WritePcf)
FON/FNT	Windows (alt)	(M1316)	FED15 (BitmapFontWriter.WriteFnt - das blanke .fnt; der NE-Behaelter .fon bleibt ein reiner Leseweg)
UFO · designspace	Quellformat der Editoren		
SFD (FontForge) · TTX (fonttools-XML)	Austausch	SFD; TTX (Metrik, Kontur, zusammengesetzte Glyphen, CFF , post)	SFD; TTX (dieselben)
AFM/PFM (Metriken)	Alter Druck		

#	Funktion	Stand bei dir	Aufw.
FED1	Glyphen-Umriss ändern - Punkte/Kurven lesen, bearbeiten und zurückschreiben (glyph und CFF-Charstrings) - M1364 (Fonts/GlyphPath = bearbeitbares Modell mit exakter Grad-Erhöhung quadratisch->kubisch und angenäherter Gegenrichtung; Fonts/GlyphOutlineEditor liest/schreibt beide Umriss-Arten; Fonts/GlyphOutlineTables = glyph/loca + Type-2-Charstrings + CFF-Tabelle). Grenzen: kubisch->quadratisch ist eine Annäherung bis zur Fehlerschranke (Vorgabe 0,5 Einheiten); ein ERSETZTER Glyph verliert Hintung und wird einfach statt zusammengesetzt (unveränderte behalten ihre Bytes); bei CFF wird die ganze Tabelle neu gebaut, also entfallen Subroutinen, Hints und die Glyphnamen - eine CID-keyed CFF wird benannt abgelehnt		L

#	Funktion	Stand bei dir	Aufw.
FED2	Neue Schrift von Grund auf: Glyphen aus Umrissen, Metriken, cmap, Namen, OS/2 - als .ttf/.otf schreiben - M1364 (Fonts/FontBuilder.BuildTrueType/BuildOpenTypeCff; .notdef wird selbsttaetig ergaenzt, die lsb folgt der echten xMin). Grenzen: cmap nur Format 4 (BMP - Format 12 ist FED5), post 3.0 ohne Glyphnamen; kein Kerning (FED4), keine GSUB/GPOS, kein Hinting, keine zusammengesetzten Glyphen, keine Variationsachsen (FED9)		L
FED3	Wandeln in jede Richtung - TrueType CFF (quadratisch kubisch umrechnen), Type1 -> OpenType, sfnt -> WOFF/WOFF2, Type42 -> sfnt (ueber das neue Kompositionsmodul FontConversion/Type42SourceReader - Fonts darf PostScript nicht importieren, dieselbe Loesung wie ImageConversion in M1305) und CFF2 -> CFF		L
FED4	Metriken und Kerning bearbeiten: Fonts/FontMetricsEditor.SetMetrics (Vorschubbreiten + linke Seitenraender je Glyphe, hmtx/hhea) und SetKerningPairs (klassische kern-Tabelle, Format 0). Grenze: geschrieben wird kern, nicht GPOS - gelesen werden beide (GlyphKerning)		M
FED5	Zeichenzuordnung bearbeiten: Fonts/FontCmapEditor.SetCmap (cmap anlegen/aendern, Format 4/6/12) und Fonts/FontPostNames.SetNames/Read (Glyphennamen, post Format 2.0)		M
FED6	Zusammenfuehren/Aufteilen: Fonts/FontGlyphSetEditor.Merge (Glyphen aus mehreren TrueType-Schriften in eine, Composite-Referenzen umgeschrieben) und ExtractSubset (Teilmenge herausloesen mit Umbenennen ueber FontPostNames und Codepunkt-Neuzuordnung ueber FontCmapEditor - mehr als das reine TrueTypeSubsetter-Subsetting). Grenze: nur glyf/loca-Schriften, gleiches UnitsPerEm, Layout-Tabellen werden beim Merge verworfen		M
FED7	Schrift-Editor im Pruefstand: Glyphe waehlen, Umriss sehen und ziehen, Metriken setzen, Vorschau, Speichern - die Oberflaeche ueber FED1-FED6 - (/schriftditor, Playground/Model/FontEditorSupport + Components/Pages/FontEditor.razor): Umriss-Punkte live per Zeiger ziehen (reine SVG-Pointer-Ereignisse auf dem SVG-Wurzelement, keine JS-Bruecke noetig), Vorschubbreite setzen, Struktur-Befund (FED8) anzeigen, mehrere Glyphen aendern und in einem Zug speichern (GlyphOutlineEditor.SetOutlines + FontMetricsEditor.SetMetrics)		L
FED8	Pruefer fuer selbst gebaute Schriften: Fonts/FontStructureChecker.Check - Pflichttabellen, Tabellen-Pruefsummen, Metrik-Widersprueche (numberOfHMetrics/hmtx-Laenge, loca/glyf-Konsistenz, cmap-GID-Bereich). Das Gegenstueck zu Extraction.FontHealthChecker (der prueft Schriften in einem fremden PDF, dieser eine eigenstaendige Schriftdatei)		M

#	Funktion	Stand bei dir	Aufw.
FED9	Variable Fonts SCHREIBEN: aus mehreren fertigen statischen Schnitten (z. B. Regular+Bold) eine variable Schrift bauen - M1384 (Fonts/VariableFontBuilder.Build + GvarWriter): fvar (Achsen + benannte Instanzen), gvar (Punkt-Deltas mit eingebettetem Peak-Tupel und aneinandergrenzenden Wirkungszonen bei mehreren Mastern auf einer Achse), avar (Umbiegung der Skala), schlanke STAT; die Achs- und Instanznamen wandern in die name-Tabelle. Nicht interpolierbare Glyphen (andere Punktzahl, andere Konturgliederung, andere Komponenten) werden benannt abgelehnt. Fremdverifiziert mit fontTools (einmalig beim Bauen, kein Python im Testlauf). Grenzen: kein IUP (die Deltas stehen fuer alle Punkte - normgerecht, nur groesser); keine Eck-Master (ein Schnitt darf hoechstens eine Achse vom Standard abweichen - das volle Variationsmodell mit sich ueberlagernden Zonen fehlt); kein HVAR/MVAR/VVAR/cvar (die Vorschubbreiten variieren ueber die Phantompunkte in gvar); nur TrueType-Umrisse (CFF2 schreiben waere ein eigener Weg); STAT ohne AxisValue-Eintraege		XL
FED1	CID-keyed CFF schreiben (ROS, FDArray/FDSelect) — die Form, in der CJK-Schriften ausgeliefert werden		L - (Fonts/CidKeyedCffWriter.MakeCidKeyed): aus einer name-keyed bare-CFF wird eine CID-keyed bare-CFF (ROS + Identitaets-Charset CID=GID + ein FDArray-Font-DICT + FDSelect Format 3 mit einem Bereich); CharStrings, Global-/lokale Subroutinen und das Private-DICT werden byteidentisch uebernommen. Ueber CffOpenTypeWrapper.Wrap in eine ladbare .otf verpackbar. Grenze: genau ein Font-DICT (keine Aufteilung auf mehrere Private-DICTs je Glyphengruppe); nur CharstringType 2; lokale Subrs muessen unmittelbar hinter dem Private-DICT liegen.
FED1	Type42 schreiben (sfnt in PostScript verpackt) — fuer die PostScript-Ausgabe		S - (Fonts/Type42Writer.Write): /sfnts traegt die vollen sfnt-Rohbytes hexadezimal (in 65534-Byte-Blocke zerlegt), /CharStrings bildet jeden Glyphnamen auf seine GID ab (der native Type-42-Mechanismus des Interpreters uebernimmt die Umriss-Interpretation). Grenze: Glyphnamen gNN (keine post-Namen-Aufloesung), /Encoding StandardEncoding; nicht gegen Ghostscript verifiziert (auf dieser Maschine nicht installiert) - strukturell geprueft ueber den Rundlauf der eingebetteten sfnt-Bytes.
FED1	Sammlungen (TTC/OTC) lesen und schreiben: mehrere Schriften mit geteilten Tabellen in einer Datei		M - (Fonts/FontCollection): Read zerlegt jede Schrift der Sammlung in ihre Tabellen, Write setzt mehrere Schriften wieder zusammen und legt byteidentische Tabellen nur einmal ab (Dedup - der Sinn des Formats). Mit fontTools (TTCollection) als unabhaengigem Orakel gegengelesen. Grenze: TTC-Kopfversion 1.0 (kein DSIG-Verzeichnis wie Version 2.0); head.checkSumAdjustment wird nicht neu gerechnet (bei geteilten Tabellen ueber mehrere Schriften nicht eindeutig auf eine Schrift beziehbar, von Betrachtern praxisnah nicht geprueft).

#	Funktion	Stand bei dir	Aufw.
FED1	Farbfonts schreiben: COLR/CPAL v0 (M1364, ColorFontEncoder) - Rest : COLR v1 (Fonts/ColorFontV1Encoder - Verlaeufer linear/radial/konisch ueber die allgemeine PaintTransform-Affine, Komposition ueber PaintComposite, PaintColLayers+LayerList wie bei v0; gemeinsamer Tabellen-Zusammenbau ColorFontEncoder.ReplaceTables mit v0) und die SVG -Tabelle (Fonts/ColorFontSvgEncoder - sortierte, ueberlappungsfreie Dokumentliste, optionales Gzip, ueberlappende Bereiche werden benannt abgelehnt statt eine fuer die Binaersuche des Lesers unbrauchbare Datei zu schreiben). Grenze: die variablen v1-Paint-Formate (M359) werden nicht geschrieben (jede Schrift ist eine statische Instanz)		L
FED1	Web-Altlasten: EOT und SVG-Fonts lesen und schreiben — beide abgeloeset, aber in Bestanden vorhanden. Fonts/EotFont+Fonts/SvgFontFile, je Leser/Schreiber. EOT: Version 0x00020001, unkomprimiert (kein MicroType-Express-Codec — benannt abgelehnt); die eingebettete sfnt-Schrift bleibt ueber den vorhandenen TrueTypeFont-Leser gueltig. SVG-Fonts: das abgeloesete -Element mit <glyph>-Pfaddaten (eigener minimaler d-freier Attributweg, kein Svg-Modul-Import — Fonts darf Svg nicht importieren)		M
FED1	Bitmapschriften schreiben: BDF, PSF1/PSF2, PCF und Windows FON/FNT (Fonts/BitmapFontWriter: WriteBdf/WritePsf/WritePcf/WriteFnt) — je gegen den vorhandenen Leser gerundlaeuft; die zwei Format-Fallen sind getestet: PCF legt die Bits je nach Kopf LSBit zuerst ab, das Windows-FNT die Punktmuster spaltenweise		M
FED1	macOS-Bestaende: dfont/Suitcase lesen (Fonts/DfontReader) und schreiben (Fonts/DfontWriter) — sfnt-Ressourcen in der klassischen Resource-Fork, mit den 24-Bit-Versaetzen. Ehrlich benannte Grenze: nur sfnt-Ressourcen, keine FOND/NFNT (die alten Bitmap-Suitcases)		M
FED1	UFO + designspace lesen und schreiben — das Quellformat, in dem Schriften entworfen werden; damit laesst sich mit anderen Editoren zusammenarbeiten. Fonts/UfoFont+Fonts/UfoSerializer (reine Bytes, wie KdocSerializer) + Fonts/UfoFolder (die einzige Stelle mit Dateisystem-Zugriff, wie KdocFolder) + Fonts/DesignSpaceFile. v1-Grenze: die gaengige fontinfo.plist-Teilmenge (nicht alle 200+ moeglichen Schluessel), kein lib.plist/Kerning/Gruppen/Komponenten-Glyphen		M

#	Funktion	Stand bei dir	Aufw.
FED1	Austauschformate: FontForge-SFD und fonttools-TTX (XML) lesen und schreiben. Fonts/SfdFont + Fonts/TtxFont. Umgesetzt: Kopfmetriken, Glyphenreihenfolge/-namen, Fortschrittsbreiten, Unicode-Zuordnung (cmap format 4), bei SFD die vollen Kontur-Umrisse (SplineSet), bei TTX die Umriss-Geometrie einfacher UND (Rest-Teil 2) zusammengesetzter glyph-Glyphen (TtxFont.Glyphs/TtxGlyphComponent, baut auf Fonts/GlyphPath auf, M1364 - kein zweiter Umriss-Weg; die implizite Mittelpunkt-Regel bei zwei aufeinanderfolgenden Kontrollpunkten ist umgesetzt und ueber eine hand-gebaute Fixture belegt; eine zusammengesetzte Glyphe traegt jetzt ihre Verweis-Bestandteile samt Versatz/Skalierung statt uebersprungen zu werden), der CFF -Umriss (Type-2-Charstrings als eigene Textform, TtxFont.CffGlyphs/Fonts/Type2CharstringText - alle Pfad-Operatoren inkl. der kompakten Formen hmoveto/hlineto/hhcurveto/vvcurveto/hvcurveto/vhcurveto werden gelesen [verifiziert an einer echten fonttools-Ausgabe], geschrieben wird ueber die vier einfachsten Operatoren + GlyphPath.ToCubic() [verlustfrei, anders als beim glyph-Schreiber keine Ablehnung]; Hint-/Unterprogramm-/Flex-Operatoren und endchar mit Akzent Aufbau werden je Glyphe uebersprungen statt geraten) und die post-Namenstabelle (Format 2.0, extraNames nach derselben Regel wie fonttools' eigener Kodierer berechnet - mit der 258-Namen-Standard-Mac-Reihenfolge als Datentabelle in Fonts/TtxFont.cs). FED18 damit komplett, keine offenen Punkte mehr.		M
FED1	Metrikdateien AFM/PFM lesen und schreiben (gehoren zu Type1). Fonts/AfmMetrics (Text, inkl. Kerning-Paare) + Fonts/PfmMetrics (binaerer GDI-Kopf + PostScript-Erweiterungskopf)		S
FED2	MCP + Pruefstand: convert_font (deckt sich mit FNT5), font_info, font_edit und ein Reiter fuer den Editor - (MCP-Werkzeuge font_info/font_edit in McpTools.All/dist/mcpb/manifest.json/McpToolGroups/Mcp.razor, dazu die eigenstaendige Pruefstand-Seite /schriftditor als FED7)		S

v4-TXT · Klartext / Notepad

#	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

#	Notepad-Funktion	Stand bei dir	Aufw
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)

#	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

#	Vorbild	Editor-Funktion	Stand bei dir	Aufw
ED30	InDesign	Zeichenformat-Palette (eigene Absatz- UND Zeichenformate anlegen/verwalten/bearbeiten)	eigene Absatz- + Zeichenformate ueber Model/EditorStyleBook + Components/Editor/StylePalette.razor (ueber Kuonen.PDFTools.Styles.StyleCatalog, kein zweiter Formatmechanismus; getrennt von den 4 festen ED6-Formaten)	M
ED31	InDesign	Ausrichten & Verteilen + Duplizieren-und-versetzen — deckt LID5	M1347 (EditorAlignDistribute/EditorStepAndRepeat + Oberflaeche: 6 Ausrichtungen an Auswahl oder Seite, Verteilen nach Zwischenraum/Mitte, Step-and-Repeat)	M
ED32	InDesign	Objekt-Effekte-Palette (weiche Kante, Schein, Relief, Satin) — deckt LID1	Model/EditorEffectsPalette + Components/Editor/EffectsPalette.razor (ueber die LID1-Effekte, M1347)	M
ED33	InDesign	Story-Editor (Fliesstext-Fenster getrennt vom Layout, fuer langen Text)	Model/EditorStoryText + Components/Editor/StoryEditor.razor (Rahmenkette am Stueck bearbeiten, Kopf bekommt den Text, Folgerahmen werden vom Ueberlauf-Mechanismus wieder gefuehrt)	M
ED34	InDesign	Verknuepfte statt eingebettete Bilder + Verknuepfungen aktualisieren/ersetzen/fehlende finden	Verknuepfungs-Palette (Model.EditorImageLinks + ImageFrame.ReplaceImageBytes)	L
ED35	InDesign	Live-Preflight-Panel im Editor (die Pruefung gibt es als Dienst, nicht als Palette)	Preflight-Palette (Model.EditorPreflightPanel, springt zur Seite)	M
ED36	InDesign	Softproof-/Ueberdrucken-Vorschau im Editor (ICC-Engine + Separationsvorschau sind da)	Softproof-Umschalter im Ansicht-Menue (Model.EditorSoftProof)	M
ED37	Acrobat	Kommentar-/Markup-Werkzeuge im Editor (Notiz, Hervorheben, Formen, Freitext) — das Annotations-Modul ist voll da	Playground/Services/EditorAcrobatSession + Components/Editor/AcrobatToolsPanel.razor, an einem PDF-Export des Editor-Dokuments (wie AP7)	M
ED38	Acrobat	Schwaerzen im Editor (RedactionEditor gibt es als Dienst, ED17)	selbe Sitzung/Feld - Markierungen sammeln, entfernen, anwenden	M
ED39	Acrobat	Signieren im Editor (SignatureEditor gibt es als Dienst, ED18)	selbe Sitzung/Feld - Zertifikat laden, Flaeche aufziehen, sofort signieren	M
ED40	Acrobat	Zwei PDFs vergleichen im Editor (PdfComparer ist da)	selbe Sitzung/Feld - zweites PDF hochladen, je Seite zusammengefasste Abweichungen	S
ED41	Acrobat	Tag-Baum + Leseordnung bearbeiten (Barrierefreiheit) — die Editor-Seite von AP1	M1373	L
ED42	Acrobat	Fremdes PDF in-place editieren (ins gerenderte PDF klicken statt Import->Re-Render) = AP7	M1373	XL
ED44	Word	Aenderungsverfolgung + Kommentare im Editor (TrackedText/EditorialNote sind da)	Playground/Model/EditorTrackChanges + Palette: Markup-Sicht (Einfuegung blau unterstrichen, Loeschung rot durchgestrichen, je mit Autor), Zaehlung, Kommentare anlegen/entfernen, Markup-PDF. Grenze des Modells: nur „alle annehmen"/„alle ablehnen" (kein selektives Annehmen); Notizen als Anhang am Dokumentende, nicht als Randnotiz	M

#	Vorbild	Editor-Funktion	Stand bei dir	Aufw
ED45	Word	Rechtschreibung beim Tippen (rote Wellenlinie) + Wort-/Zeichenzahl in der Statusleiste	Playground/Model/EditorProofing: der Text wird in Stuecke zerlegt (nichts geht verloren), unbekannte Woerter tragen die Wellenlinie; die Zaehlung steht im Fliesstext-Feld und in der Editor-Statuszeile. Grenze: die eingebaute Wortliste des SpellChecker ist kompakt - ohne Morphologie meldet sie auch gewoehnliche Woerter; eigene Woerter lassen sich mitgeben. Keine Grammatikpruefung, keine Korrekturvorschlaege	M
ED46	Word	Kopf-/Fusszeilen im Editor bearbeiten (der Stempel-Dienst HeaderFooterStamper ist da)	Playground/Model/EditorHeaderFooter + Palette: sechs Zonen, Groesse/Rand/erste Seitenzahl/Deckblaetter/Datumsformat, und eine Vorschau je Seite, die zeigt, was {page}/{pages}/{date} dort wirklich ergeben. Grenzen des Stempels: die Zeile liegt im Seitenrand (der Satzspiegel schrumpft nicht), je Zone eine Zeile ohne Auszeichnung, keine getrennten linken/rechten Seiten	M
ED47	Excel	Tabellenblatt-Modus: Zellgitter, Formeleingabe (auch UEBER Blattgrenzen, Blatt2!A1), mehrere Blaetter	SpreadsheetPanel (Components/Editor/) - aus dem grossen Editor heraus erreichbar (Werkzeugleiste -> „Tabellenblatt...“) UND weiter als eigenstaendige Seite /tabellenblatt; XlsxFormula.EvaluateAcrossSheets loest Blatt2!A1 auf (jedes Blatt wird zuerst fuer sich neu berechnet, ein unbekanntes Blatt wird als #FEHLER gemeldet, nie still 0). Ehrlich benannte v1-Grenze: nur Einzelzell-Bezuege - ein blattuebergreifender BEREICH (Blatt2!A1:B2) wird benannt abgelehnt	L
ED48	PowerPoint	Folien-Modus (Foliensortierung, Layouts, Platzhalter)	SlideSorterPanel (Components/Editor/) - aus dem grossen Editor heraus erreichbar (Werkzeugleiste -> „Folien...“) UND weiter als eigenstaendige Seite /folien; SlideSorter.UpdateShapeGeometry/UpdateShapeText bearbeiten Stelle/Groesse/Text eines Rahmens (Platzhalters) direkt in der Oberflaeche. Ehrlich benannte v1-Grenze: ein Kind einer Gruppe ist ueber diesen Weg nicht einzeln erreichbar	L
ED49	PowerPoint	Praesentieren (Vollbild-Vorfuehrung, Referentenansicht, Notizen)	als eigenstaendige Pruefstand-Seite /praesentieren (PresenterView: Navigation + Notizen der RICHTIGEN Folie; Vollbild ueber die Fullscreen-API, aktuelle+naechste Folie als Referentenansicht)	M
ED50	Dreamweaver	HTML/CSS-Code-Editor mit Live-Vorschau (Split-View CodeAnsicht, /htmleditor) - jede Eingabe wandelt sofort ueber denselben HtmlConverter wie html_to_pdf		L
ED51	Dreamweaver	Projekt-/Site-Verwaltung (Dateibaum, lokale src/href-Verweise gegen den Projektordner pruefen, SiteFileTree/SiteLinkChecker)		M
ED52	Dreamweaver	Code-Vervollstaendigung + Validierung (HTML/CSS)	HTML: Model/HtmlTagCompletion + HtmlValidator (nicht geschlossene und falsch verschachtelte Elemente). CSS: Model/CssEditorSupport (Eigenschafts- und Wert-Vervollstaendigung, unbekannte Eigenschaft, nicht geschlossener Block) gegen die EINE Liste Html/CssPropertyCatalog im Kern - mit Drift-Waechter, der sie gegen den Quelltext des HtmlStyleResolver haelt, damit Vorschlag und tatsaechliches Koennen nicht auseinanderlaufen	M
ED53	Notepad	Klartext-Tab im Editor (siehe TXT1)	Components/Editor/PlainTextEditorPanel - der Klartext-Kern aus M1320 (Kodierung, Zeilenenden, Suchen/Ersetzen mit Zeile+Spalte, Gehe-zu-Zeile, Syntax, Klammerpaare) jetzt auch als Werkzeug im grossen Editor, kein zweiter Klartext-Weg gegenueber /texteditor	M

v4-FS · Datei- und Ordner-Werkzeuge (Nutzerwunsch)

#	Punkt	Stand	Aufw
FS1	file_info — alles, was der Eigenschaften-Dialog zeigt: Groesse, alle drei Zeitstempel (erstellt / geaendert / zugegriffen), die Attribute (schreibgeschuetzt, versteckt, System, Archiv, komprimiert, verschluesselt), ob es eine Verknuepfung / Abzweigung ist und wohin sie zeigt, und die vollstaendige Versionsressource: Dateibeschriftung, Datei- und Produktversion, Produktname, Firma, Copyright, Originaldateiname und Sprache. Gemessen an einer echten Systemdatei: alle acht Felder kommen ueber FileVersionInfo eingebaut zurueck, wortgleich zum Windows-Dialog	M1424: file_info liefert alle acht Felder der Versionsressource, die drei Zeitstempel, die sechs Attribute und das Ziel einer Verknuepfung. Nicht dabei, je mit Grund: die Groesse auf dem Datentraeger (GetCompressedFileSize), die Dateiarart im Klartext samt "Oeffnen mit" (steht in der Registrierung, nicht in der Datei) und die Vorgaengerversionen (Schattenkopien)	M
FS2	Attribute und Zeitstempel setzen — schreibgeschuetzt / versteckt / System / Archiv setzen und loeschen, Zeitstempel aendern; einzeln oder rekursiv	M1424: set_attributes und set_timestamps, einzeln oder rekursiv, ueber den geteilten Lauf (FS7). Steht der Wert schon so da, meldet der Bericht schon richtig statt geaendert - bei tausend Dateien ist genau das die Zahl, die zaehlt. Versteckt/System/Archiv gibt es unter Unix nicht: dort benannt, nicht nachgebildet (ein fuehrender Punkt waere eine Umbenennung, also ein ganz anderer Eingriff)	S
FS3	Berechtigungen LESEN — unter Windows die ACL (je Eintrag Konto, Rechte, Erlauben/Verweigern, geerbt ja/nein), unter Linux und macOS der POSIX-Modus samt Besitzer und Gruppe. Nie geraten: auf dem jeweils anderen System wird die fehlende Form benannt, nicht nachgebildet	M1424: file_permissions - Windows Besitzer, Gruppe und jeder Zugriffseintrag (Konto, Rechte, erlaubt/verweigert, geerbt); Unix der POSIX-Modus. Die auf dem System fehlende Form wird gesagt, nie nachgebildet. Nicht dabei: die Pruefungsliste (SACL/Auditierung, Privileg SeSecurity), und unter Unix Besitzer und Gruppe - die stat-Struktur liegt je System und Architektur anders, sie zu raten waere schlimmer als sie wegzulassen	M
FS4	Berechtigungen SETZEN — Rechte hinzufuegen, entfernen, ersetzen; Vererbung ein- und ausschalten; einzeln oder rekursiv. Mit Trockenlauf als Vorgabe: ein rekursiver Rechte-Eingriff ist der gefaehrlichste Aufruf im ganzen Werkzeugkasten — er zeigt erst, was er tun wuerde, und tut es erst auf ausdrueckliche Anweisung	M1424: set_permissions - hinzufuegen, entfernen, ersetzen, Vererbung ein-/ausschalten, einzeln oder rekursiv. Trockenlauf ist die Vorgabe (erst apply=true fuehrt aus), und es braucht zwei Freigaben zugleich: KUONEN_PDFTOOLS_ENABLE_FS_WRITE=1 UND erhoehte Rechte - das ist strenger als Windows selbst (dort darf ein Nutzer die Rechte an einer eigenen Datei aendern) und ausdruecklicher Nutzerwunsch	L
FS5	Besitzer aendern — einzeln oder rekursiv. Ehrlich vorweg: unter Windows braucht das das Privileg SeTakeOwnership (also erhoehte Rechte), unter Unix in aller Regel root; und einen verwalteten chown gibt es in .NET nicht — das laeuft ueber DllImport("libc"), wie es SpoolerStart.AktuelleKennung (M1414) fuer geteuid schon tut. Fehlt das Recht, muss das Werkzeug es sagen, statt still nichts zu tun	M1424: set_owner, gleiche Schranke und gleicher Trockenlauf. Windows nennt bei fehlendem Recht das Privileg SeTakeOwnership beim Namen. Unter Unix laeuft es ueber DllImport("libc") und nimmt Nummern (uid bzw. uid:gid): einen Namen aufzuloesen braeuchte dieselbe nicht verwaltete passwd-Struktur, deshalb wird ein Name abgelehnt statt geraten	M

#	Punkt	Stand	Aufw
FS6	Symbole aus beliebigen Dateien	M1424/M1425: file_icon deckt die drei Wege (Symboldatei .ico/.cur/.icns, eingebettetes Symbol einer Ressourcendatei, Zuordnungssymbol einer Dateart aus der Registrierung), entschieden an der Signatur, nicht an der Endung. M1425 hat den Rest geschlossen: der Store-Paket-Verweis (@{Paket?ms-resource://...}, gemessen bei .txt/.png/.jpg/.mp3) wird ueber SHLoadIndirectString von der Shell aufgeloeset - zurueck kommt ein echtes, lesbares PNG unterhalb von WindowsApps; die negative Zahl ist eine Ressourcen-ID und wird ueber die neu herausgegebenen Gruppen-IDs gesucht (imageres.dll, -68 = das Symbol von .bat), nicht als Position geraten; und den ausgelagerten .mun-Ressourcen wird gefolgt (System32\imageres.dll ist ein 12-KB-Rumpf, die 24 MB liegen daneben - ohne diesen Schritt liefe der ganze Weg fuer Systemdateien ins Leere). Grenzen, die Windows setzt: der Paket-Verweis liefert eine Fassung (die der laufenden Sitzung), nicht alle im Paket abgelegten Groessen; ausserhalb von Windows gibt es weder Zuordnung noch Paket-Verweis und beides wird benannt abgelehnt	M
FS7	Rekursion, Filter und Bericht als durchgaengiges Prinzip — jede aendernde Aktion nimmt dieselben Angaben (Muster, Tiefe, nur Dateien / nur Ordner) und liefert je Eintrag, was geschah: geaendert, schon richtig, uebersprungen, fehlgeschlagen mit Grund. Ein stiller Sammel-Lauf ueber tausend Dateien waere wertlos	M1424: DateiLauf - Muster, Tiefe und Art fuer jede aendernde Aktion, dazu der Bericht je Eintrag (geaendert / schon richtig / uebersprungen / fehlgeschlagen mit Grund). Die Vorgabe ist der sichere Fall: Tiefe 0 = nur der genannte Pfad; das Muster filtert die Kinder, nie den ausdrecklich genannten Pfad. Ein Wurf im Einzelfall wird zum Eintrag, nicht zum Abbruch	M
FS8	Signatur einer Programmdatei lesen — der Reiter „Digitale Signaturen“: wer signiert hat, mit welchem Verfahren, wann gegengezeichnet. Gemessen: den Unterzeichner liefert .NET eingebaut (CN=Microsoft Corporation kam an Ihrer Datei zurueck). Ehrlich die Grenze: das ist das blosses Zertifikat — die Pruefung (Kette bis zur Wurzel, Zeitstempel, Sperrliste, stimmt der Dateiinhalt noch zur Signatur) ist etwas anderes und braucht WinVerifyTrust ueber P/Invoke. Ein Werkzeug, das „signiert von X“ meldet, ohne geprueft zu haben, waere gefaehrlicher als keines	M1424: file_signature liest Antragsteller, Aussteller, Gueltigkeit, Verfahren und Fingerabdruck. Die Grenze ist der Punkt: nichts davon ist geprueft - weder Vertrauenskette noch Zeitstempel noch Sperrliste, und nicht, ob der Dateiinhalt noch zur Signatur passt; das braeuchte WinVerifyTrust. Der Bericht sagt es in jeder Ausgabe. Gemessen: die Dateien in System32 sind katalogsigniert, tragen also gar keine eingebettete Signatur - daraus "nicht signiert" zu machen waere eine falsche Aussage ueber eine signierte Datei	M

INF · Betrieb und Auslieferung (Nutzerwunsch)

#	Punkt	Stand
INF5	Ein Dienst darf fehlen duerfen: der Pruefstand zeigt einen Reiter nur, wenn die zugehoerige Instanz wirklich antwortet	fuer den Spooler (M1400: Lebendmeldung /gesundheit, kurzes Zeitlimit, Tab verschwindet sonst) · fuer den Noten-Dienst

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

#	Funktion	Stand bei dir	Aufw
MUS:	Notenmodell: Partitur -> Stimmen -> Takte -> Ebenen -> Noten/Pausen; Tonhoehe, Dauer, Vorzeichen, Schluessel, Tonart, Taktart, Haltebogen, Balken, N-tolen, Vorschlagsnoten	Music (Score/Part/Measure/Voice/Note/MusicRest, dazu Pitch/NoteDuration/Clef/KeySignature/TimeSignature/TupletSpan). Ein reines Datenmodell - es rechnet und zeichnet nichts, wie QrBill und Invoicing. Stammton und Versetzung bleiben getrennt, sonst waeren cis und des nicht mehr unterscheidbar	M
MUS:	Stech-Regeln (der Kern): Notenkopf-Positionen, Halsrichtung und -laenge, Balkengruppen und ihre Neigung, Hilfslinien, Vorzeichen-Stapelung, Kollisionsvermeidung und die horizontale Verteilung	Music/ScoreEngraver + EngravingRules. Kopfposition aus Schluessel und Stufe (vier Schluessel), Halslaenge eine Oktave, Balken bis Zweiunddreissigstel mit begrenzter Neigung und Teilbalken, Hilfslinien, Vorzeichen-Stapelung, Ausweichen der Sekunde im Akkord, dauerabhanger Abstand, Punktierungen, Fahnen, verkleinerte Vorschlagsnoten. Seit M1404: mehrere Stimmen je Takt auf derselben Zeile mit Halsrichtung nach der Stimmenrolle (nicht nach der Lage - daran unterscheidet sich echter Mehrstimmensatz von „zweimal einstimmig uebereinander“), Haltebogen ueber Taktstrich und Zeilenumbruch, Instrument-Vorgabeschluessel und Tonumfangspruefung je Takt. Offen, je gemeldet: Bindeboegen, N-tolen-Klammer, Halsverlaengerung zur Mittellinie, Balkenlage auf Linienmitte	XL
MUS:	MusicXML lesen (.musicxml und gepackt .mxl) — der Weg herein aus jedem anderen Programm	MusicXml/MusicXmlReader (M1404), neues Blatt-Modul mit der einen Kante auf Music. Beide Erscheinungsformen (die ZIP-Signatur entscheidet, nicht die Endung), part-list, Schluessel/Tonart/Taktart, Tonhoehe mit Stammtone und Versetzung, Dauer mit Punktierung, Pausen, Akkorde, Haltebogen, Balken, Vorzeichen, Vorschlagsnoten, N-tolen. Die zwei Fallen des Formats sind je mit Gegenbeleg getestet: duration steht in divisions (je Datei frei gewaehlt, nicht in Vierteln), und backup setzt die Zeit im Takt zurueck. XXE-fest. Gegen music21 verifiziert. Offen: score-timewise wird benannt abgelehnt; Liedtext/Dynamik/Boegen/Wiederholungen werden gemeldet, nie still verschluckt	L
MUS:	MusicXML schreiben — der Weg hinaus; damit ist der Rundlauf zu, wie bei DOCX/XLSX	MusicXml/MusicXmlWriter (M1404) - unkomprimiertes score-partwise 4.0, Teiler 3360 (so gewaehlt, dass jede Dauer des Modells ganzzahlig wird), backup zwischen mehreren Stimmen. music21 liest die eigene Ausgabe deckungsgleich zum Original zurueck, inklusive zweier gleichzeitiger Stimmen im selben Takt. Offen: die gepackte .mxl wird gelesen, aber nicht geschrieben	M
MUS:	MIDI lesen und schreiben (Standard MIDI File 0/1) samt Anbindung an das Notenmodell	Der Codec (Midi, M1372) und die Bruecke zur Notation (MidiNotation, M1404): Note-An/Note-Aus paaren, quantisieren (Massstab ist die Kopf-Aufloesung, nicht eine feste Tickzahl - daran faellt ein naiver Wandler auf), Notenwerte mit Punktierung, Takte aus dem Taktart-Meta-Ereignis, Enharmonik aus der Tonart (MIDI 61 ist cis oder des - geraten wird nie), Stimmen je Spur/Kanal, Schluessel aus dem Tonumfang. Gegen mido verifiziert. Offen, je gemeldet: kein Haltebogen ueber den Taktstrich (der Ton beginnt den naechsten Takt ganz und klingt damit spaeter als gespielt), nur eine Stimme je Takt, keine N-tolen, kein Tempo und keine Dynamik; SMPTE-Zeitcode wird benannt abgelehnt	M

#	Funktion	Stand bei dir	Aufw
MUS	MCP-Werkzeuge und ein Pruefstand-Reiter	(M1404) midi_to_pdf, musicxml_to_pdf, midi_to_musicxml und musicxml_info - Letzteres sagt vor dem Wandeln, was verlorengehen wird. midi_to_pdf und musicxml_to_pdf geben beide Berichte aus (Uebersetzung und Notenstich); sonst bliebe unsichtbar, dass ein Ton wegen des fehlenden Haltebogens spaeter klingt als gespielt. Damit ist die Fehlerklasse geschlossen, die hier dreimal auftrat (IMG1/IMG22/IMG27): gebaut, aber nicht ansteuerbar	S

MUS13 · Der Noten-Editor im Einzelnen — 5 von 14

#	Scheibe	Stand
MUS13.1	Eigene Seite /noteneditor mit eigenem Knopf neben dem Editor	M1403
MUS13.2	Mehrere Spuren mit Instrument je Spur (39 Instrumente), Systemgruppen	M1404
MUS13.3	Rechenkern MusicEditing (diatonisch verschieben, Zeitwert, Takteinteilung) — rein und testbar, wie TableGridEditing	M1403
MUS13.4	Vorschau auf den bedruckten Bereich zugeschnitten (eine A4-Seite ist zu ueber 90 % weiss)	M1403
MUS13.5	Import: MusicXML, MIDI, PDF und jedes Bildformat	M1404

v4-PRN · Virtueller PDF-Drucker (Nutzerwunsch)

#	Drucker-Funktion	Stand bei dir	Aufw
PRN	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/IppPrintService (netzfreier Kern) + Playground/IppPrintServer (POST /ipp/print). Seit M1391 zusaetzlich Create-Job/Send-Document (RFC 8011 §4.2.2/§4.2.3) - die zweiphasige Einreichung, bei der Cancel-Job zwischen den zwei Anfragen wirklich greift (der Auftrag steht dazwischen als Pending, noch ohne Dokument/Verarbeitung); Print-Job bleibt einphasig-synchron und damit unveraendert praktisch nie abbrechbar (dafuer nutzt ein Client den zweiphasigen Weg). Ausserdem opt-in ipps:// (IppTlsCertificate + KUONEN_PDFTOOLS_IPPS_PORT/KUONEN_PDFTOOLS_TLS_CERT_PATH) und die bereits vorhandene globale Basic-Auth-Schranke (AUTH1) deckt /ipp/print wie jeden anderen Pfad. Bekanntmachung im Netz siehe PRN3. Seit M1392 zusaetzlich Get-Job-Attributes (RFC 8011 §4.3.4 - die Auskunft ueber EINEN Auftrag) sowie die volle Auftragssteuerung Hold-Job/Release-Job/Restart-Job/Purge-Jobs (RFC 8011 §4.3.6-§4.3.9, neu IppJobState.Held + IppPrintQueue.Hold/Release/Restart/Purge, bewusst enge Uebergangs-Waechter: Hold nur aus Pending, Restart nur aus Completed) - job-state-reasons traegt fuer einen zurueckgestellten Auftrag jetzt job-hold-until-specified (aus dem vorhandenen Zustandsmodell abgeleitet, nicht erfunden). Pause-Printer/Resume-Printer bleiben ehrlich unimplementiert (der Dienst arbeitet synchron, kein Zustand „angehalten“)	L

#	Drucker-Funktion	Stand bei dir	Aufw
PRN2	Spooler + Oberfläche: Warteschlange mit Absender/Zeit/Seitenzahl/Status, Vorschau, Herunterladen, Loeschen, Aufbewahrungsdauer	Printing/PrintSpoolRetention (reine Ablaufregel, ohne Uhr) + Playground/Services/PrintSpoolService (baut auf dem Ausgabeordner auf, den IppPrintServer bereits fuellt - IppPrintQueue hat weder Persistenz noch eine Loeschfunktion) + Seite /druckerwarteschlange. Seit M1391 zeigt die Warteschlange zusaetzlich die ECHTE Herkunft (Absender-Adresse + optional requesting-user-name, aus einer Sidecar-Datei neben der Auftrags-PDF, Playground/Model/PrintSpoolOrigin) - Absender bleibt der frei wahlbare job-name, Herkunft ist die verlaessliche Angabe	M
PRN4	Rueckgabe an das ausloesende Geraet: Link/QR-Code direkt nach dem Druck, wahlweise Ablage in einem ueberwachten Ordner oder Netzlaufwerk	Playground/Model/PrintSpoolReturnLink (Link + QR-Code ueber die vorhandenen Barcode-Bausteine, ueber den Barcode-Leser rueckgelesen geprueft) + PrintSpoolService.TryDropToWatchedFolder (Kopie in einen konfigurierbaren Ordner). Grenze: ein Netzlaufwerk muss bereits als lokaler Pfad erreichbar sein (kein UNC-Mount mit eigenen Zugangsdaten)	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	Printing/PrinterPpd (PPD aus IppPrinterAttributes abgeleitet - keine Widersprueche zwischen PPD und IPP-Selbstbeschreibung) + Playground/Services/PostScriptDropFolder (opt-in ueber KUONEN_PDFTOOLS_PS_DROPFOLDER, wandelt ueber den vorhandenen PostScriptToPdf). Einrichtung in docs/drucker-einrichten.md. Grenze: gegen einen echten Windows-Treiber auf dieser Maschine nicht pruefbar (kein Windows-Drucksystem im Testlauf) - geprueft ist die PPD-Syntax gegen die Adobe-Spezifikation	M

#	Drucker-Funktion	Stand bei dir	Aufw
---	------------------	---------------	------

PRN3 · Der Drucker findet sich von selbst — 7 von 7

#	Scheibe	Stand
PRN3	mDNS-Datensätze bauen (_ipp._tcp, AirPrint-TXT-Schlüssel) — netzfreier Kern	M1390 - von dnspython gegengelesen
PRN3	UDP-Multicast-Versand im Spooler	M1390
PRN3	Die Bekanntmachung muss den Container verlassen	M1409 - gemessen, nicht beschrieben (Lauscher auf dem Wirt, mit unserem eigenen Mdns-Leser). Positivkontrolle ohne Docker: 3 Pakete. Bruecken-Netz: nichts (kuendigt 172.17.0.2 an). network_mode: host: nichts - unter Docker Desktop ist das „Wirtsnetz“ die VM (192.168.65.6), nicht das LAN. macvlan: nichts, die Adresse war vom Wirt nicht erreichbar. Wirtsseitiger Ankuendiger: 4 Pakete, Adresse erreichbar. WSD (239.255.255.250:3702) verhaelt sich gleich: aus dem Container nichts, vom Wirt 5 von 5. Folge: Linux-Server -> Wirtsnetz oder macvlan (erwartet, mangels Maschine nicht messbar); Windows-/macOS-Wirt -> wirtsseitiger Ankuendiger (scripts/netzweg-ankuendiger.cs, baut die Pakete aus Mdns/IppServiceAnnouncement - kein zweiter Paketbau). Ein mDNS-Reflektor wurde bewusst NICHT gebaut: er spiegelt das Paket, aber die Adresse darin bleibt 172.17.0.2 - das Geraet faende den Drucker und kaerne nicht an ihn heran
PRN3	IPP Everywhere vollstaendig (PWG 5100.14): printer-device-id, ipp-features-supported, media-col-database, printer-icons, printer-strings-uri	M1409 - das ist der „Treiber“: bei IPP Everywhere gibt es keinen, das System baut die Warteschlange aus der Selbstauskunft. Neu die IPP-Sammlungen nach RFC 8010 §3.1.6 (schreiben und lesen - ohne sie liesse sich media-col-database gar nicht ausdruecken), die Pflichtangaben samt Deckungsnachweis, sowie Geraetesymbol und Sprachdatei gerechnet statt mitgeliefert (kein Binaerbestand im Repo). Ehrlich offen: ipptool vermisst weiterhin printer-make-and-model, printer-info, printer-location, printer-more-info, printer-up-time und media-col-default
PRN3	Probing, Konfliktbehandlung und Abmelde-Pakete (RFC 6762 §8.1/§10.1)	M1409 - MdnsProbing (dreimal sondieren im Abstand von 250 ms, Vorschlag im Authority-Abschnitt), MdnsRecordOrder (der lexikografische Vergleich aus §8.2.1 - ohne Cache-Flush-Bit, ohne Lebensdauer, Namen kleingeschrieben), MdnsGoodbye (§10.1) und MdnsResponder (Bekannte-Antwort-Unterdrueckung §7.1, QU-Bit §5.4). Der Fund beim Bauen: die erste Fassung wertete JEDEN Datensatz als Konflikt - ein PTR auf _ipp._tcp.local ist aber ein geteilter Eintrag, ein zweiter Drucker dort ist der Normalfall; der Drucker haette sich im erstbesten Netz mit einem zweiten endlos umbenannt. Jetzt zaehlen nur die eindeutigen Datensaeetze. Offen: die Drosselung nach §9 (>15 Konflikte in 10 s), der Konflikt im laufenden Betrieb, die Zufallsverzoeigerung 20-120 ms fuer geteilte Datensaeetze (§6) und die Fortsetzung einer Known-Answer-Liste ueber Pakete (§7.2)



#	Scheibe	Stand
PRN3	Windows: WSD (Web Services on Devices) als zweiter Weg	M1409 - neues Blatt-Modul Wsd (importiert nichts, wie Mdns): alle sechs Nachrichten (Hello/Probe/ProbeMatches/Resolve/ResolveMatches/Bye) schreiben und lesen, Gerätebeschreibung, WS-MetadateExchange, Antwortregel mit APP_MAX_DELAY 500 ms (Zeit und Zufall werden hineingereicht). Fassung 2005/04, nicht die spätere OASIS-Fassung - Windows spricht bis heute die alte. Die Gerätekennung wird aus dem Rechnernamen abgeleitet, nicht gewuerfelt: sonst legte Windows bei jedem Neustart einen neuen Drucker an. Fremdverifiziert gegen die Python-Bibliothek WSDDiscovery in BEIDEN Richtungen (ihre Nachrichten als Fixture-Konstanten im Repo, kein Fremdwerkzeug im Testlauf), quergelesen an wsdd. Der unterscheidende Fall: die fremde Probe nennt den Druckertyp unter dem Praefix zzz statt wprt - wer Typen samt Praefix als Zeichenkette vergleicht, antwortet einem echten Client nie. Ehrlich benannte Grenze: WSD ist hier nur der Erkennungsweg. WS-Print (die Druckseite mit CreatePrintJob/SendDocument) ist NICHT gebaut - gedruckt wird ueber IPP, und die Metadaten verweisen dorthin. Ebenfalls offen: IPv6, WS-Security, WS-Eventing

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

#	Lightroom-Funktion	Stand bei dir	Aufw
LR2	Grundeinstellungen: Belichtung, Kontrast, Lichter, Tiefen, Weiss, Schwarz, Klarheit, Dunst entfernen, Dynamik, Saettigung	Photo/BasicAdjustments (M1404). Belichtung als Multiplikation in linearem Licht (auf den sRGB-Zahlen gerechnet saehe es bei kleinen Werten fast richtig aus - genau das war einer der gefundenen Fehler); die vier Tonwertbereiche ueber eine Maske aus der wahrgenommenen Luminanz (mittleres Grau liegt wahrgenommen bei 0,50, linear bei 0,216 - mit der linearen Maske zog „Tiefen“ bis weit in die Mitteltoene); Weissabgleich in Mired statt linear in Kelvin (linear wird der Rot-Faktor jenseits ~18 300 K negativ). Offen: Kontrast dreht um einen festen Drehpunkt (kein S-Kurven-Modell), Klarheit ist Unscharf-Maskieren mit Box-Filter, Dunst ohne Guided-Filter-Verfeinerung	L
LR3	Tonwertkurve (Punkt + parametrisch je Kanal), HSL/Farbmischer, Farbkalibrierung	Photo/ToneCurve + Photo/ColorMixer (M1404). Die Kurve ist monoton (die uebernommene Fassung schwang ueber und erzeugte bei gewoehnlichen Stuetzpunkten an 16 Stellen eine echte Tonwertumkehr), alle vier parametrischen Regler wirken ueber ihre ganze Bahn, und die acht Farbbereiche sitzen an den echten Farbtonstellen (gleichmaessig alle 45 Grad verteilt bekam ein reines Blau von seinem Regler nur zwei Drittel). Neutrale Bildpunkte fallen nicht mehr ins Rot-Band. Offen: die Farbkalibrierung; die Zugehoerigkeit haengt allein am Farbton, nicht zusaetzlich an Helligkeit/Saettigung	L
LR6	Transformieren: Perspektive/Ausrichten (Upright), Zuschchnitt, Drehung, Spiegeln	Photo/GeometryTransform (M1404): Drehen um Vielfache von 90 Grad, Spiegeln waagerecht und senkrecht, Zuschneiden (ein Bereich ausserhalb wird benannt abgelehnt, nie still geklemmt), Geraderichten um einen beliebigen Winkel und Perspektivkorrektur. Ein Punkt ausserhalb wird durchsichtig, nie geraten. Offen: das automatische Upright (die Erkennung der Fluchtlinien), und Straighten behaelt die Bildmasse - die Ecken werden abgeschnitten	M

#	Lightroom-Funktion	Stand bei dir	Aufw
LR8	Effekte: Vignettierung nach Zuschnitt, Koernung	Photo/PhotoEffects (M1404) - radiale Abdunklung mit quadratischer Kennlinie (die Kantenmitte trennt sie von der linearen; Ecke und Mitte waeren bei beiden gleich) und achromatisches Filmkorn (alle drei Kanale aendern sich um denselben Betrag)	S

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

Punkt	Stand	Was noch fehlt / warum
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.

INF · Betrieb und Auslieferung (Nutzerwunsch)

Wie die Bausteine als Container zusammenkommen. Seit M1412 ist der Wunsch umgesetzt: drei Dienste — Pruefstand, Noten und Druck-Spooler — laufen je als eigener Container, jeder ist dreifach erreichbar (allein und in beiden Standard-Verbuenden), und alle drei kommen aus einem Bild. Dazu kam als Unterbau das Werkzeugbild, das die Fremdwerkzeuge traegt.

#	Punkt	Stand
INF1	Noten als eigener Dienst — alles rund um Noten (Notensatz, MusicXML/MIDI, OMR, Noten-Editor) laeuft als eigene Instanz, auf die zugegriffen wird	M1412 (Playground/Services/DienstProfil + KUONEN_PDFTOOLS_DIENST=musik): eigener Container, eigene Adresse, eigene Oberflaeche - aber dasselbe Bild, dieselbe Entscheidung wie beim Spooler. Browser-belegt in beide Richtungen (mit der Angabe: / -> /noteneditor, genau ein Reiter; ohne sie: 25 Reiter). Ein unbekannter Dienstname faellt auf den vollen Pruefstand zurueck und wird benannt. Ehrliche Grenze: der Dienst schaltet die Oberflaeche um, nicht die Bibliothek - wer eine andere Adresse eintippt, bekommt sie (Gliederung, kein Zugangsschutz)
INF2	Der Noten-Dienst in allen drei Compose-Dateien: docker-compose.yml, docker-compose.tools.yml und eigenstaendig als docker-compose.musik.yml	M1412 - alle drei; ein Waechter haelt sie zusammen (DienstVerbundTests)
INF3	Derselbe Dreiklang fuer den Spooler: allein (docker-compose.spooler.yml, besteht seit M1400) und zusaetzlich in den beiden Standard-Dateien	M1412 - in beiden Standard-Verbuenden; der Pruefstand ist dort Kunde des eigenstaendigen Spoolers (KUONEN_PDFTOOLS_SPOOLER_URL), startet also keinen zweiten. Als einziger Dienst mit veroeffentlichtem Host-Port (631, dem genormten IPP-Port) - ein Drucker, den kein Geraet erreicht, ist keiner
INF4	Auslieferung ueber GitHub nur mit docker-compose.yml und docker-compose.tools.yml	M1412 - so ausgelegt (und in docker/README.md als Auslegung benannt): die uebrigen Compose-Dateien bleiben im Repo, aber weder rebuild.sh noch die Veroeffentlichungs-Kette ruehren sie an. Die zwei ausgelieferten tragen jetzt alle drei Dienste

#	Punkt	Stand
INF6	Der Waechter zieht mit: TesseractImageTests haelt heute fest, dass es genau zwei Bilder gibt — kommen eigene Dienste dazu, muss er das mit Begrueundung abbilden statt umgangen zu werden	M1412 - es bleibt bei zwei Anwendungsbildern: die neuen Dienste sind Container desselben Bildes, kein drittes. Dazu kommt das Werkzeugbild als Unterbau, und der Waechter belegt die Begrueundung statt sie zu behaupten: es traegt keine Zeile Anwendungscode (kein ENTRYPOINT, kein COPY --from=build)
INF7	Der Bau fuellt den Cache-Speicher, statt ihn zu nutzen (vom Nutzer gemeldet: 9,22 von 10 GB, ueber hundert Eintraege)	M1412 - Ursache gemessen: der GitHub-Cache ist nach Referenz abgeschottet, der Deploy laeuft aber auf einen Tag - jeder Tag eine eigene Referenz. Was ein Tag-Bau schrieb, konnte der naechste nie lesen: Schreibmuell, der unter dem 10-GB-Deckel ausgerechnet den einzigen brauchbaren Cache (den von main) verdraengte. Dagegen drei Dinge: die teuren Fremdwerkzeuge liegen jetzt als eigenes Bild auf Docker Hub (kein Deckel, keine Verdraengung, gebaut nur bei neuem Schluessel aus tool-versions.env + Rezept, mit der .NET-Fassung im Namen); ein Tag-Bau schreibt keinen Cache mehr (er liest den von main); und ein Aufraeum-Workflow loescht unlesbare Eintraege und haelt den Rest unter Budget. Ehrlich offen: die Wirkung zeigt sich erst am naechsten Bau - belegt ist die Ursache, nicht der Erfolg
INF8	Der Docker-Bau brach ab (vom Nutzer gemeldet: dotnet publish Exit 1)	M1412 - .dockerignore schloss docs/ aus und holte eine Datei zurueck; M1411 hatte die Roadmap in fuenf Teildateien aufgeteilt und eingebettet. Im Bau-Kontext fehlten sie, lokal faellt das nie auf. Am echten Docker belegt: mit der Korrektur 6 Dateien im Kontext, ohne sie 1. Ein Waechter haelt csproj und .dockerignore jetzt zusammen - dieselbe Fehlerklasse wie IMG1/IMG22/IMG27

Warum „eigene Instanz“ nicht automatisch „eigenes Bild“ heisst: der Spooler ist seit M1400 ein zweiter Einstiegspunkt desselben Bildes — zwei Container, ein Bild. Das war die kleinere Aenderung und haelt beide Dienste zwangslaeufig auf demselben Quellstand. Fuer den Noten-Dienst ist dieselbe Frage bewusst zu entscheiden, nicht nebenbei.

IMG6 · AVIF lesen (AV1-Intra-Decoder) — abgeschlossen mit M1413

#	Punkt	Stand	
IMG6 8/8	AVIF lesen (AV1-Intra-Decoder) — in Scheiben wie der HEVC-Decoder (M1111–M1147)	Untertabelle unten: IMG6.1–6.8	XL

#	Scheibe	Stand
IMG6	Nachfilter (Deblocking, CDEF, Loop Restoration) + YUV->RGB, PixelsDecoded = true	M1413 - alle drei in Normreihenfolge zwischen Rekonstruktion und RGB, samt get_source_sample (§7.17.6: im Streifen die Werte nach CDEF, darüber/darunter davor) und den drei noetigen Bildfassungen. Gegen dav1d auf den rohen Farbebenen: 17 Dateien, 0 Abweichungen - darunter eine, die vorher 73223 von 73728 Bildpunkten daneben lag. Dabei fielen zwei still weggeworfene Rahmenkopf-Angaben auf (loop_filter_delta_enabled/ref_deltas) und ein latenter int-Ueberlauf in Av1TileGroup.Split. Filmkorn (§7.18.3) bleibt bewusst draussen: Absicht-Rauschen, kein Bildinhalt

v4-LPP · PowerPoint — in M1420 geschlossen

#	PowerPoint-Funktion	Stand	Aufw
LPP1	Layouts + Master + Platzhalter (sldLayout/sldMaster/ph)	M1420: Lesen (Vererbung sldMaster -> sldLayout -> Folie) UND Schreiben - eine Form, die ihre Lage vom Platzhalter erbt, wird auch wieder ohne eigenes <a:xfrm> geschrieben; ein Schreiber, der die Lage einbetoniert, kaeme durch einen blossen Positionsvergleich durch. Noch nicht gebaut: die Platzhalter aller Layouts landen in einem geschriebenen Layout	L
LPP3	Formen-Geometrie (prstGeom) + Fuellung/Kontur/Verlauf/Schatten	M1420: Lesen war vollstaendig, Schreiben jetzt ebenso - jede bekannte Geometrie (Ellipse, RoundRect, vier Pfeile, Stern, Callout), gradFill, Kontur mit Breite, effectLst/outerShdw	L
LPP4	Aufzaehlungen + Absatzformat (buChar/buAutoNum, Einzuege, Zeilenabstand, Tabulatoren)	M1420: gelesen UND geschrieben (lvl, marL, lnSpc, tabLst in der OOXML-Abfolge); die Tabulator-Sprungpositionen werden jetzt wirklich angewandt (Vorgabe ein Zoll, wo keine festgelegt sind). Noch nicht gebaut: die Ausrichtung einer Tabulator-Position (algn)	M

v4-MUS · Notensatz und Noten-Editor — in M1421 geschlossen (M1426: MUS2 gebuendelt, MUS3 feine Stichregeln)

#	Punkt	Stand	Aufw
MUS2	Notenschrift anbinden und buendeln: SMuFL-Glyphennamen -> Codepunkte -> Glyph-Nummern, Referenzschrift Bravura (SIL OFL 1.1) in Core ab Werk gebuendelt	M1426: Music/BundledMusicFont + SmuflMusicFont.Bundled. Offizielle Steinberg Bravura.otf als eingebettete Ressource in Kuonen.PDFTools.Core gebuendelt. MusicPdf.Render, ScoreEngraver, MCP und Noten-Editor nutzen Bravura ab Werk als Vorgabe - Schluessel (Violin/Bass/Bratsche), Vorzeichen, Pausen, Notenkoepfe und Faehnchen werden ohne externe Schriftanforderung vollstaendig und sauber gesetzt. Gezielte Vektor-Unterdrueckung ueber SuppressMusicFont moeglich	S

#	Punkt	Stand	Aufw
MUS3	Die feinen Stichregeln: Balkenneigung, Balkenraster (Sit/Straddle/Hang), Ausweichen bei Kollisionen (Bindeboegen, Fermaten, Beischriften), elastischer optischer Abstand	M1426: EngravingRules + ScoreEngraver + ScoreEngraver.Expression. Melodische Konturanalyse fuer waagerechte Balken bei Wiederholungen und Wechselnoten, Einrasten von Balkenneigung an Notenlinien (Sit, Straddle, Hang), Halsverlaengerung bis zur Mittellinie fuer Noten auf Hilfslinien, einheitliche Halsseite in Balkengruppen, dynamisches vertikales Ausweichen von Beischriften (Dynamik, Gabeln, Silben, Akkorde) und Fermaten, dynamische Woelbungsanpassung von Bindeboegen zur Umgehung dazwischenliegender Notenkoepfe, elastische Spaltenbreitenkompensation fuer gestaffelte Vorzeichen und lange Silben	XL
MUS5	Alles am Notentext: Dynamik, Artikulation, Bindeboegen, Triller/Verzierungen, Oktavierungen, Pedal, Fermaten, Atemzeichen, Tempo- und Vortragsangaben	M1421: Music/MusicExpression + ScoreEngraver.Expression - je ins Modell, in den Notentext und ins PDF. Die Stichregeln sind eingehalten: Artikulation auf der halsabgewandten Seite, der Bindebogen woelbt sich entgegen der Halsrichtung, und er ist strukturell vom Haltebogen getrennt (ein Haltebogen verbindet dieselbe Tonhoehe, ein Bindebogen verschiedene)	L
MUS6	Text an den Noten: Gesangstext (mehrere Strophen, Silbentrennung, Melismen), Akkordsymbole, Fingersatz, Generalbass	M1421. Ein stiller Fehler dabei behoben: DrawTexts legte sich einen eigenen Bericht an und warf ihn weg - ein nicht setzbares Zeichen in einer Silbe wurde brav bereinigt und die Meldung darueber an niemanden gerichtet; der Aufrufer sah einen stillschweigend gekuerzten Gesangstext	M
MUS13	Freistehend wie der grosse Editor: Werkzeugleiste, Eigenschaften-Feld, Rueckgaengig-Kette, Speichern/Laden im Arbeitsbereich	M1421 - MusicEditorState haelt Dokument und Verlauf als reine Funktionen (jede Aenderung ergibt ein neues Dokument), gespeichert wird ueber den vorhandenen MusicXml-Weg im Arbeitsbereich	
MUS13	Auswahl und Ziehen: einzeln, Rahmenauswahl, mehrere; verschieben in Tonhoehe UND Zeit	M1421 - MusicSelection nach dem Vorbild von EditorMarquee, Verschieben ueber MusicCursorEditing. Die Anker kommen aus der gesetzten Partitur, damit es nur eine Takteinteilung gibt	
MUS13	Klaviatur unten, ein- und ausblendbar; eine Taste setzt die Note an der aktuellen Stelle	- die Klaviatur laesst sich ein- und ausblenden, eine schwarze Taste ist ein erhoelter Ton (ihr Vorzeichen schlaegt das der Palette); MusicKeymap.TonhoeheAusMidi rechnet eine MIDI-Nummer in eine Tonhoehe, die Wahl cis/des trifft der Aufrufer statt zu raten	
MUS13	Eingabe ueber die Rechnertastatur (Tonhoehe, Notenwert, Vorzeichen, Weiterruecken)	M1421 - a-g setzen, die Pfeiltasten verschieben die Auswahl in Tonhoehe und Zeit, Entf loescht; browser-verifiziert am laufenden Pruefstand	

v4-IMG · Nachtrag — zwei Zeilen, die in der Restliste haengeengeblieben waren

Gefunden in M1422 auf Nutzerfrage. Beide waren gebaut; die alte Statusregel konnte sie nur nicht sehen (bei IMG22 ueberstimmte ein der Beschreibung das der Statusspalte, bei IMG17 war nach M1400 schlicht die Marke nicht nachgezogen).

#	Punkt	Stand	Aufw
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. + M1400: die Ratensteuerung (PCRD-opt) - eine echte Zielgroesse in Bytes bzw. ein Kompressionsverhaeltnis, ueber die konvexe Huelle im Rate-Distortion-Raum und einen Lagrange-Multiplikator. Ueber ImageConvertOptions und convert_image (target_bytes/compression_ratio) erreichbar	L
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

v4-ARC · Archive packen und entpacken — in M1433 geschlossen (alle Punkte)

#	Format / Funktion	Lesen	Schreiben	Au
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	M1426: ZIP64 (>4 GB, >65535 Eintraege), WinZip AES (AE-1/AE-2, 128/192/256 Bit), ZipCrypto, Deflate/Store/Bzip2/LZMA/Zstd, Kommentare, Unix-Rechte, erweiterte Zeitstempel (0x5455/0x000A)	M1426: ZIP64-EOCD/Locator, WinZip AES-128/192/256, ZipCrypto, Deflate/Store/LZMA, Unix-Rechte (0644/0755), erweiterte Zeitstempel (0x5455), Kommentare	M
ARC2	BZIP2 (.bz2) — Burrows-Wheeler, Huffman	M1423: eigener Dekodierer (Huffman mit Tabellenwechsel, MTF, inverse BWT, RLE), byte-genau gegen Pythons bz2	der Kodierer ist ein eigenes Stueck Arbeit (Sortierung aller zyklischen Verschiebungen)	M
ARC3	TAR-Ketten: .tar.gz, .tar.bz2, .tar.xz, .tar.zst als EINE Einheit behandeln, dazu die Unix-Rechte/Besitzer/Symlinks	M1427: .tar.gz, .tar.bz2, .tar.xz und .tar.zst als EINE Einheit erkannt und gelesen; Unix-Rechte (0755/0644/0600), Besitzer/Gruppen, UID/GID und Symlinks in ArchiveEntry und TarArchiveWriter	.tar.gz und .tar mit konfigurierbaren Unix-Rechten und Besitzern geschrieben (TarWriteOptions)	S
ARC4	XZ / LZMA / LZMA2 lesen (.xz, .lzma)	M1426: XzDecoder, Lzma2Decoder und LzmaCodec.Entpacke samt Bereichsdekodierer, Fenster und Modell	nur LZMA (LzmaCodec.Packe, LzmaRangeEncoder) - LZMA2 und XZ schreiben ist ARC3b	L
ARC5	Zstandard entpacken (.zst)	M1426: ZstdDecoder mit FSE- und Huffman-Tabellen	— der Packer ist ARC5b	L
ARC6	LZ4 (.lz4) — schnell, in Linux-Werkzeugketten verbreitet	M1426: Lz4BlockCodec und Lz4FrameCodec, Rahmen-Pruefsumme ueber die eigenen XxHash32/XxHash64	M1426: Block- und Rahmenformat, die Block-Endregeln je einzeln belegt	S

#	Format / Funktion	Lesen	Schreiben	Au
ARC3	compress (.Z) — das alte Unix-Format	M1426: CompressLzwCodec (LZW mit wachsender Codebreite, Blockmodus)	M1426: CompressLzwCodec.Encode	M
ARC3	LZMA2 und XZ schreiben	siehe ARC3a	M1433: Lzma2Encoder (Kontrollbyte, gepackte/ungepackte Bloecke bis 64 KiB; Woerterbuch-Reset und Eigenschaften-Uebermittlung als ZWEI Fahnen - ein frueher ungepackter Block liess die Eigenschaften sonst ungesagt) und XzEncoder (Stream-Kopf/-Fuss, Index, CRC32/CRC64/SHA-256)	M
ARC4	7z (.7z): LZMA/LZMA2, BCJ, feste Blockbildung, Verschluesselung	M1433: Signaturkopf, gepackter Hauptkopf, Verfahrensketten Speichern/LZMA/LZMA2/BCJ-x86/Delta/AES-256 (die Reihenfolge aus den Bindepaaren, nicht aus der Kopf-Reihenfolge - py7zr schreibt sie je nach Kette verschieden), solide Bloecke; gegen py7zr und 7-Zip	M1433: Speichern/LZMA/LZMA2, BCJ-x86, solide, gepackter Kopf; AES-Schreiben nicht	L
ARC5	Zstandard packen	siehe ARC5a	M1433: Rahmen mit XXH64, Bloecke roh/RLE/gepackt, Huffman-Literale, drei FSE-Tabellen, Hash-Kette ueber Blockgrenzen - und DREI Fehler im Leser, die erst zstd zeigte (FSE-Huffman-Gewichte um eins zu viel, kanonische Codes falsch sortiert)	L
ARC8	RAR entpacken (RAR4 und RAR5)	M1433: beide Behaelter, RAR-2.9- und RAR-5.0-Verfahren samt Filtern (Standard-VM-Filter am Fingerabdruck erkannt); RAR5-Codelaengen absolut statt als Unterschied (an einem echten Mehrblock-Archiv gefunden). Nicht: mehrteilige Archive, Wiederherstellungsdaten, verschluesselte RAR	nie — die RAR-Lizenz untersagt einen Kompressor; rechtliche, keine technische Grenze	L
ARC9	CAB (MSZIP/LZX)	M1433: Store, MSZIP und ein eigener LZX-Dekodierer; der Ordner-Strom laeuft ueber Dateien UND CFDATA-Bloecke hinweg und wird je Ordner einmal ausgepackt (wer je Block neu beginnt, liefert ab Block 2 Muell)	M1433: Store und MSZIP (kein LZX-Packer)	M
ARC10	Apple: .dmg (HFS+/APFS) und Apple Archive (.aar/.yaa)	M1433: UDIF mit Stueckarten roh/Null/ADC/zlib/bzip2/LZFSE/LZMA, Partitionssuche (direkt/GPT/Apple Partition Map), HFS+/HFSX samt Links und decmpfs, APFS (Pruefpunkt, Objektkarten, Fletcher-64, decmpfs mit zlib/LZVN/LZFSE), eigener LZFSE/LZVN-Entpacker, Apple Archive blank und im pbz*-Mantel; gegen libmodi, libfshfs, libfsapfs, liblzfse/libfmos. Nicht: verschluesselte/versiegelte/Fusion-APFS, LZBITMAP, bvx1, Pruefsummen im .dmg, das HFS-Journal	nicht gebaut	L

#	Format / Funktion	Lesen	Schreiben	Au
ARC:	ISO 9660 / UDF	M1433: ISO mit Joliet und Rock Ridge (CE-Fortsetzungen, rr_moved, mehrteilige Dateien), UDF 1.02-2.60 mit allen vier Partitionsarten (physisch, Sparable, VAT, Metadaten-Partition); gegen xorriso/genisoimage/mkudffs, bsdtar und den Linux-Kern. Nicht: verschraenkte Dateien, Mehrsitzung, ZF, benannte Stroeme	M1433: ISO 9660 mit Joliet und Rock Ridge, byte-gleich wiederholbar, von bsdtar/xorriso/isoinfo/7z/Kern gelesen (Fortsetzungsflaechen direkt hinter dem Verzeichnis, sonst verliert libarchive lange Namen). Nicht: UDF schreiben, El Torito, Dateien ueber 4 GiB	M
ARC:	Stuffit, cpio, ar	M1433: cpio (newc/crc/odc/binaer, Hardlinks), ar (GNU/BSD, Symboltabelle, .deb), Stuffit klassisch und 5 mit den Methoden 0/1/2/3/13/15; gegen GNU cpio/ar, bsdtar, dpkg-deb,unar/lsar. Nicht: Stuffit-Verschluesselung, Methoden 5/6/8/14, Stuffit X	M1433: cpio (newc) und ar; Stuffit nicht (kein freier Packer zum Gegenpruefen)	M
ARC:	Fassade + MCP + Pruefstand	M1423/M1433: ArchiveDetector/ArchiveReader/ArchiveWriter ueber ALLE Formate, archive_list/archive_extract/archive_create (Gruppe archive), der Reiter /archive	M1433: archive_create und der Reiter packen zip/tar/Ketten/7z/cab/cpio/ar/iso und die Ein-Datei-Mantel	S

Ehrlich zu RAR: entpacken ist nachgebaut; packen nicht und nie — die RAR-Lizenz untersagt, aus den veroeffentlichten Unterlagen einen RAR-Kompressor zu bauen. Allen Formaten gemeinsam: das Archiv liegt ganz im Speicher (fuer sehr grosse Abbilder braeuchte es einen stroemenden Weg).

v4-LIC · Lizenz-Manager — in M1427 geschlossen (7 von 7)

„ich moechte das der lizenz maenager endlich detailliert ist aber gruppen oder einzel auswahl moeglich ist“

Was heute da ist, am Code nachgesehen. Licensing/LicensedFeatures kennt neun grobe Merkmale — gpu.acceleration, batch.processing, commercial.use, security.removal, pdf.creation, instance.full, show.downloads, hide.extras, signature.export — plus license.unlimited. Eine Lizenz schaltet damit ganze Themenbloecke frei oder nicht; ein einzelnes Werkzeug laesst sich gar nicht adressieren.

Der tragende Befund: die halbe Arbeit ist schon getan. Die Einteilung existiert

(Mcp/McpToolGroups: 13 Themen ueber 192 Werkzeuge — genau die Gliederung, die der Pruefstand unter „Werkzeuge im Ueberblick“ zeigt), und McpToolListing.ForList filtert bereits danach (M1244/M1267: KUONEN_PDFTOOLS_BLOCKED_GROUPS nimmt gesperrte Gruppen aus tools/list UND aus dem Aufruf). Was fehlt, ist nicht der Filter, sondern die Quelle der Entscheidung: heute ist es der Schalter des Betreibers, kuenftig zusaetzlich das Recht aus der Lizenz. Es darf kein zweiter Filter entstehen — sonst hat die Anwendung zwei Stellen, an denen ein Werkzeug verschwinden kann, und niemand findet mehr heraus, welche gegriffen hat.

#	Punkt	Stand	Aufw
LIC1	Das Lizenzmodell wird fein: neben den neun groben Merkmalen traegt eine Lizenz eine Liste freigegebener Gruppen (AllowedGroups) und eine Liste einzelner Werkzeuge (AllowedTools). Abwaertskompatibel: eine Lizenz ohne diese Listen wirkt wie bisher — sonst faellt beim ersten Einspielen jedem Bestandskunden die Haelfte weg		M
LIC2	Eine einzige Stelle entscheidet — die Lizenz wirkt durch denselben McpToolListing.ForList/IsBlocked-Weg (McpToolPolicy.CheckAccess) wie die Sperrliste, nicht daneben. Die Entscheidung liefert dabei den Grund mit (Allowed, NotLicensed, Blocked, Unknown), damit alle Faelle praezise unterscheidbar bleiben		M
LIC3	Der Lizenz-Manager als Oberflaeche — in der Darstellung der Werkzeug-Uebersicht: 13 Gruppen mit Zaehler, aufklappbar, Haken auf der Gruppe ODER auf dem einzelnen Werkzeug, „alle/keine“ je Gruppe, Suche ueber alle 204. Eine halb angehakte Gruppe muss als solche zu sehen sein	M1427: /lizenzmanager - 17 Gruppen mit Zaehler, aufklappbar, Haken auf Gruppe ODER Einzelwerkzeug, Suche ueber alle 204; die halb angehakte Gruppe ist als solche zu sehen (orange, halbgefuelltes Quadrat). Rechenkern Playground/Model/LizenzManager (reine Funktionen). Der Manager stellt keine Lizenz aus - er gibt die Mengen zum Kopieren aus	L
LIC4	Dieselbe Auswahl beim Ausstellen — LicenseGen und LicenseAdmin schneiden eine Lizenz mit genau derselben Auswahl zu, statt nur die groben Merkmale zu kennen (--groups/--tools im CLI bereits aktiv)	M1427: geteilte Pruefung Licensing/LicenseSelectionCheck im Kern - beide Aussteller rufen sie; ein unbekannter Name wird benannt abgelehnt (Rueckgabewert 5, keine Datei), dazu --blocked (war fest auf null verdrahtet) und die Auflistung dessen, was die Lizenz wirklich freigibt. --features bleibt ungeprueft (Kennungen sind bewusst offen)	M
LIC5	Sichtbar statt still — ein nicht lizenziertes Werkzeug verschwindet aus tools/list und nennt beim Aufruf den Grund samt dem, was die Lizenz traegt (McpToolNotLicensedException meldet Werkzeug, Gruppe, Lizenznehmer sowie freigegebene Gruppen und Einzelwerkzeuge). Ein Werkzeug, das wortlos fehlt, schickt den Nutzer auf die falsche Fehlersuche		S
LIC6	Der Pruefstand zieht mit — Reiter und Seiten folgen derselben Lizenz wie die MCP-Werkzeuge; heute haengt die Oberflaeche an den groben Merkmalen (show.downloads, hide.extras)	M1427: Playground/Model/SeitenLizenz + PlaygroundSecurity.SeiteErlaubt, durchgesetzt an einer Stelle um @Body (eine Schranke je Seite liefe auseinander). Abwaertskompatibel: eine Lizenz ohne Gruppen-/Werkzeugangaben verhaelt sich genau wie bisher, und eine Seite ohne Sachgebiet wird nie gesperrt. Sichtbarkeits-Regel, kein Zugangsschutz - die Faehigkeiten selbst schuetzen die Feature-Gates und die MCP-Durchsetzung	M

#	Punkt	Stand	Aufw
LIC7	Der Nachweis — eine Lizenz, die genau eine Gruppe traegt, gibt genau deren Werkzeuge frei und kein einziges daneben; dazu der Gegenbeleg (ohne die Gruppe ist keines davon da) und der Fall „Gruppe gesperrt, aber ein Einzelwerkzeug daraus ausdruecklich lizenziert“ (McpToolGroupLicensingTests)		M

Eine Falle, die der Zuschnitt mitbringt und die vorab benannt gehoert. Eine Lizenz, die Werkzeugnamen aufzaehlt, bindet sich an diese Namen: wird ein Werkzeug umbenannt oder aufgeteilt, verliert eine bestehende Lizenz es stillschweigend. Das ist kein Grund, es zu lassen — aber es braucht entweder stabile Kennungen neben dem Anzeigenamen oder eine gepflegte Umbenennungs-Tabelle, und einen Waechter, der beides zusammenhaelt (wie der M559-Drift-Waechter es fuer die Manifeste tut).

Und eine Abwaegung, die zum Modell gehoert. Je feiner eine Lizenz schneidet, desto mehr Stellen gibt es, an denen die Durchsetzung umgangen werden kann. Genau deshalb ist LIC2 kein Beiwerk, sondern der Kern: eine Stelle, die entscheidet, und ein Aufruf, der den Grund nennt.