Zen Working Group
Request for Discussion and Implementation Record

ZDS 0012
Format Record

The Legacy Binary Office Readers

STATE
committed

CREATED
2026-08-07

LAST UPDATED
2026-08-08

INTENDED STATUS
Committed

AUTHORS
Zen Contributors <team@insan.ai>

DISCUSSION

Mapping, omissions, and round-trip expectations for the DOC, XLS, PPT, and XLSB readers

LABELS
formats, doc, xls, ppt, xlsb, reader

Status of This Memo

This document is an internal Zen discussion record authored in Typst and tracked in git. It intentionally follows RFC-style structure so the design scope, rationale, trade-offs, and operational constraints remain explicit. Documents that still use the placeholder number XXXXX are provisional drafts. Numbered ZDS documents are part of the project record.

Contents

97. Abstract

One format record for the four legacy binary Office readers: `zenfmt_doc` (`ai.insan.zenfmt.doc`, `.doc`), `zenfmt_xls` (`ai.insan.zenfmt.xls`, `.xls`), `zenfmt_ppt` (`ai.insan.zenfmt.ppt`, `.ppt`, `.pps`, `.pot`), and `zenfmt_xlsb` (`ai.insan.zenfmt.xlsb`, `.xlsb`). The first three share the `zenfmt_cfb` compound-file (MS-CFB) container library; XLSB is the binary sibling of XLSX inside the same OPC ZIP package. These formats exist so that decades of archived documents convert without a detour through Microsoft Office; the record states exactly how much of each format is projected and what is refused.

98. Scope

`zenfmt_cfb` reads the Compound File Binary format the way `zenfmt_ooxml` reads ZIP: directory first, streams on demand, every FAT, mini-FAT, DIFAT, and directory chain walk bounded by the sector count so a crafted cycle is a refusal rather than a hang, and every stream read counted against the byte limits of ZDS 0002. The version-3 and version-4 sector sizes are both accepted; the red-black directory tree is deliberately scanned linearly — the tree shape is a hint, not a trust boundary.

The readers are text-fidelity subsets, not full decoders:

- **DOC** parses the FIB, the piece table (CLX/Pcdt) from the 0Table/1Table stream, and both piece encodings (compressed cp1252 and UTF-16LE), emitting paragraphs, hard breaks, and HYPERLINK fields as links. The STSH stylesheet and the PlcfBtePapx/PAPX bin table are resolved to map Heading 1–9 styled paragraphs (by built-in identifier or name) to headings; a malformed stylesheet degrades to plain paragraphs with a note.
- **XLS** parses BIFF8: BOUNDSHEET, SST (with CONTINUE splits and their restated encoding flags), FORMAT/XF number formats, DATEMODE, and the cell records LABELSST, LABEL, NUMBER, RK, MULRK, BOOLERR, and FORMULA with cached values (following STRING records for cached strings).
- **PPT** walks the PowerPoint Document record tree iteratively and harvests `TextHeaderAtom` + `TextCharsAtom`/`TextBytesAtom` runs in stream order.
- **XLSB** parses the MS-XLSB record framing (1–2 byte type, 1–4 byte length varint) for the workbook, shared strings, styles, and worksheet parts, resolving sheets through the workbook relationships part.

99. Mapping

99.1. DOC

Facets (ZDS 0013): headings carry a `ProvenanceFacet` naming the plugin id, the `WordDocument` stream, and the paragraph's byte position in it, with confidence `exact`. Any paragraph whose STSH style resolves to a named, non-default style carries a `StyleFacet` with that name; the default style stays unnamed on purpose.

Source	Tree result	Facets
Paragraph mark (`0x0D`)	`paragraph` boundary.	`StyleFacet` with the resolved STSH style name, for named non-default styles.
Heading 1–9 styled paragraph (STSH + PAPX istd)	`heading` with the style's level, clamped to 6.	`ProvenanceFacet`: plugin, `WordDocument` member, byte position, confidence `exact`; plus the `StyleFacet` above.

Source	Tree result	Facets
Vertical tab (`0x0B`)	`hard_break` within the paragraph.	none.
Tab (`0x09`)	A space.	none.
HYPERLINK field (`0x13`/`0x14`/`0x15`)	`link` around the field result text; `\l` targets become fragment links. Other fields keep their result text and lose their markers.	none.
Non-breaking hyphen (`0x1E`)	A hyphen.	none.

99.2. XLS and XLSB

The same projection as the XLSX reader (ZDS 0005): each sheet becomes a level-2 heading with the sheet name followed by one `table`; the first present row is the `table_head`. XLS honors the 1904 date epoch (DATEMODE); RK and MULRK integer and div-100 encodings are decoded exactly.

Source	Tree result	Facets
LABELSST, LABEL, and formula STRING results	The cell's text, shared strings resolved.	`GridFacet` with value type `text`, the record's own row and column, and the text cached.
NUMBER, RK, MULRK records, and cached formula numbers	ISO dates for date formats (builtin ids and custom format strings), percentages multiplied out, other numbers verbatim.	`GridFacet` with value type `number` or `date`; the rendered value is cached. BIFF formula source is not decoded, so `formula` stays empty.
BOOLERR records and cached formula booleans and errors	`TRUE`/`FALSE`, or the `#REF!`-	`GridFacet` with value

Source	Tree result	Facets
	style error spelling.	type `boolean` or `error_value`.
Formulas	Never evaluated — the cached value is used; `xls.formula-without-cached-value` / `xlsb.formula-without-cached-value` when there is none.	Facet coordinates are the records' row, so sparse sheets keep exact positions even where the projected table compacts them.

99.3. PPT

Source	Tree result	Facets
`TextHeaderAtom` type title or centerTitle	Level-2 heading per `\r`-separated line.	`ProvenanceFacet` on the heading: plugin, `PowerPoint Document` member, confidence `exact`.
Body and other text atoms	`paragraph` per `\r`-separated line; `0x0B` is a `hard_break`.	`ProvenanceFacet` as above.
Notes-typed text	`container` with class `notes`.	none.

100. Deliberate omissions

Construct	Report code	Reason
DOC non-heading paragraph and character styles	`doc.styles-omitted`	Heading styles are resolved through STSH and PAPX; other paragraph

Construct	Report code	Reason
		styles and the CHPX character layer are not. The note fires only when the stylesheet itself cannot be read.
DOC table grid (`0x07` marks)	`doc.tables-flattened`	Cell boundaries survive as paragraphs; the grid needs PAPX `fInTable` resolution.
DOC pictures and OLE objects	`doc.embedded-objects-dropped`	Object streams are for-

Construct	Report code	Reason
		mat soup; the text notes their presence.
DOC page breaks	`doc.page-breaks-dropped`	Markdown has no pages.
PPT positioning, images, shapes	`ppt.presentation-projection`	Same unconditional warning as PPTX: a deck loses the most.
XLS and XLSB formulas	`xls.formula-without-cached-value`	zenfmt never evaluates; cached results are used when present.
BIFF5 and older	`xls.unsupported-biff`	Pre-1997 BIFF is a

Construct	Report code	Reason
		different record set; the refusal names the fix.

101. Round-trip expectations

These are one-way readers. A round trip through Markdown keeps: full text in reading order, paragraph boundaries, hyperlinks (DOC), sheet names and cell values with their date and percent renderings (XLS, XLSB), and title, body, and notes structure (PPT). It never recovers: character styling (the legacy CHPX layer is not read), the table grid (DOC), slide layout, or any embedded object.

102. Security

- Encryption is a refusal, never an attempt: `doc.encryption-refused` (FIB `fEncrypted`), `xls.encryption-refused` (FILEPASS), `ppt.encryption-refused` (a cryptography session record).
- Every CFB chain walk (FAT, mini-FAT, DIFAT, directory) is bounded by the file's sector count; cyclic FATs and lying stream sizes are `Malformed` refusals, covered by tests.
- Stream sizes are checked against `max_entry_uncompressed` and the running total against `max_total_uncompressed` before copying.
- XLSB inherits the ZIP rules of ZDS 0002 unchanged: central directory only, hostile names reject, stored and deflate only.
- The XLSB record-length varint is capped at four bytes and clipped to the part size, so a lying length cannot run past the buffer.

103. References

- ZDS 0002, the architecture record (container rules, limits, reports).
- ZDS 0005, the XLSX reader record (the spreadsheet projection mirrored here).
- MS-CFB, MS-DOC, MS-XLS, MS-PPT, MS-XLSB (Microsoft Open Specifications).