GemBox Support Center

PdfName is decoded as UTF-8 only: Latin-1 umlauts become U+FFFD. Please add a PDFDocEncoding fallback and/or raw-byte access like in PdfString Completed

- TM Thomas M.
- **Forum name:** #GemBox.Pdf Feature Request

When a PDF stores a name object using PDFDocEncoding / Latin-1 bytes, GemBox.Pdf decodes the name as UTF-8 unconditionally. Bytes that are not valid UTF-8 are replaced with the Unicode replacement character U+FFFD, and the original byte is then unrecoverable through the public API. This affects both the raw PdfName.ToString() and the typed name-valued properties such as PdfCheckBoxField.ExportValue and PdfField.Value. PdfString falls back to PDFDocEncoding when the bytes are not UTF-16BE, and exposes its raw bytes (Length, Bytes[] indexer, enumerator). PdfName offers neither, so there is no lossless way to read such a name. Sample file: G0561_2025_10_14_Verdienstausfall_Antrag.pdf. The checkbox group AW_1 has three options whose export values / appearance-state names contain German umlauts. In the file these are stored as PDF name objects with hex-escaped PDFDocEncoding bytes (#E4 = byte 0xE4 = ä): 2613 0 obj

Comment (1)

MG **Mario - GemBox**
3 months ago

Hi Thomas,

What version of GemBox.Pdf are you using?

Please note that starting with version 2025.9.103, we extended the `PdfName.ToString()` behavior. If UTF-8 decoding is unsuccessful, it falls back to PDFDocEncoding.

Could you please try again with that version or any newer release?

If the problem persists, please send us a small Visual Studio project that reproduces the issue so we can investigate further.

Finally, please note that Feature Requests are not intended for reporting issues or asking questions. Instead, please submit a support ticket:

https://support.gemboxsoftware.com/new-ticket

Regards,

Mario