GemBox Support Center

Support updating INCLUDETEXT fields New

- 

  MRN
- **Forum name:** #GemBox.Document Feature Request

Including text fields is a part of MS Word that allows us to reuse common components. If this feature is used, it is not able to be updated via GemBox.Document. I am not familiar with a workaround for this issue.

According to this link below you claim to be releasing more updatable field types. Is there a timeline for IncludeText fields?
https://www.gemboxsoftware.com/document/docs/GemBox.Document.Field.html#GemBox_Document_Field_Update

Comment (1)

Mario - GemBox

3 years ago

Hi, Currently, the problem is that GemBox.Document supports only inline-level elements inside the Field, see the content model diagram: https://www.gemboxsoftware.com/document/docs/content-model.html However, note that we do plan to eventually add support for block-level elements inside the Field as well, see the following feature request: https://support.gemboxsoftware.com/community/view/enhance-gembox-document-content-model After that is done, we'll be able to easily add support for updating INCLUDETEXT fields. For now as a workaround, perhaps you could use something like this: var document = DocumentModel.Load("input.docx"); foreach (var field in document.GetChildElements(true, ElementType.Field) .Cast() .Where(f => f.FieldType == FieldType.IncludeText) .Reverse()) { string includePath = field.GetInstructionText().Trim('"'); var includeDocument = DocumentModel.Load(includePath); bool firstSection = true; ContentPosition position = field.Content.End; foreach (var section in includeDocument.Sections) { position = position.InsertRange(firstSection ? section.Blocks.Content : section.Content); firstSection = false; } var parent = field.Parent; var parentCollection = field.ParentCollection;

parentCollection.Remove(field); if (parentCollection.Count == 0) parent.Content.Delete(); } document.Save("output.docx"); document.Save("output.pdf"); I hope this helps. Regards, Mario