PDFBOX-4999

mkl

2020-10-23

Contents

Chapter 1

Root issue PDFBOX-4999

1.1 Summary

Dangerous COSDictionary.addAll(COSDictionary) method

1.2 Description

The method {{COSDictionary.addAll(COSDictionary)}} creates the impression, by name and by JavaDoc comment,

```
/**
 * This will add all of the dictionaries keys/values to this dictionary.
...
```

that it can be used for exactly that, adding all key/value pairs from the argument dictionary to the current one, replacing old entries for the same keys.
If one looks at the implementation, though, one is in for a surprise:

```
/**
 * This will add all of the dictionaries keys/values to this dictionary.
 * Only called when adding keys to a trailer that already exists.
 *
 * @param dic The dictionaries to get the keys from.
 */
public void addAll(COSDictionary dic)
{
    dic.forEach((key, value) ->
    {
        /*
         * If we're at a second trailer, we have a linearized pdf file, me
aning that the first Size entry represents
         * all of the objects so we don't need to grab the second.
         */
        if (!COSName.SIZE.equals(key) || !items.containsKey(COSName.SIZE))
        {
            setItem(key, value);
        }
    });
}
```

Here existing *Size* entries explicitly are not replaced!

This appears to be a relic from times when PDFBox parsed PDF documents front to back, ignoring cross reference streams, for improved results with linearized files when merging trailer dictionaries.

Nowadays this exceptional treatment of *Size* does not make any sense anymore, see [this stack overflow answer|https://stackoverflow.com/a/64502740/1729265].

Furthermore, this method is used in other contexts than creating trailer unions, even some PDFBox methods use it to create arbitrary dictionary unions:
* {{org.apache.pdfbox.pdmodel.PDDocument.assignAcroFormDefaultResource(PDAcroForm, COSDictionary)}}
* {{org.apache.pdfbox.filter.JPXFilter.decode(InputStream, OutputStream, COSDictionary, int, DecodeOptions)}}
* {{org.apache.pdfbox.examples.interactive.form.FieldTriggers.main(String[])}}
* {{org.apache.pdfbox.pdmodel.graphics.image.PDImageXObject.PDImageXObject(PDStream, PDResources)}}
* {{org.apache.pdfbox.pdmodel.graphics.image.PDInlineImage.PDInlineImage(COSDictionary, byte[], PDResources)}}
* {{org.apache.pdfbox.pdmodel.graphics.image.PDInlineImageTest.testInlineImage()}}
* {{org.apache.pdfbox.pdfparser.XrefTrailerResolver.setStartxref(long)}}

(This list is offered by eclipse as callers of that method. There may be other, hidden calls.)

Thus, this exception should be removed after all usages of that method in PDFBox have been analyzed.

1.3 Attachments

No attachments

1.4 Comments

1. **lehmi:** That's an interesting observation, never realized that. One call is missing in that list

   * {{org.apache.pdfbox.pdfparser.XrefTrailerResolver.setStartxref(long)}}

2. **mkl:** {quote}One call is missing in that list{quote}

   Indeed, I've just added it to the list.

   Originally I skipped it because I considered it to be the one case referred to in the method comment _"Only called when adding keys to a trailer that already exists."_ But that's wrong, that comment referred to the old, not xref-oriented document loading, not to the current document loading.

3. **jira-bot:** Commit 1884086 from lehmi@apache.org in branch 'pdfbox/trunk'
   [https://svn.apache.org/r1884086]

PDFBOX-4999: don't skip size values when merging COSDictionaries as proposed by Michael Klink

4. **jira-bot:** Commit 1884087 from lehmi@apache.org in branch 'pdfbox/branches/2.0'
   [https://svn.apache.org/r1884087]

   PDFBOX-4999: don't skip size values when merging COSDictionaries as proposed by Michael Klink

5. **lehmi:** The size value isn't skipped any more when merging COSDictionaries. Now, the size value is overwritten when merging the trailer dictionaries of a pdf, but that isn't an issue for two reasons. The size value isn't use after merging and the merged value is as wrong/right as the other two ones of the non merged dictionaries.

   [~mkl] Thanks for the report