LibreOffice Conference Milan, 2022

Chasing an Interoperability Bug in Impress

Sarper Akdemir

Consultant Software Engineer Intern

Collabora Productivity

sarper.akdemir@collabora.com
@quwex

LibreOffice Conference Milan, 2022

who am i

Sarper Akdemir (quwex)

- GSoC 2020, Physics Based Animation Effects
- Senior Electronics and Communication Eng. Student at Istanbul Technical University
- Served as chair of ITU Software Freedom Club two terms.
- Intern at Collabora Productivity

Collabora Productivity

LibreOffice Conference Milan, 2022

Rough steps for fixing an interoperability bug

- Understanding the problem/bug
- Coming up with a proposed fix
- Implementing the actual fix
- Implement tests for the fix

Collabora Productivity

LibreOffice Conference Milan, 2022

Import Bug (PPTX)

Collabora Productivity

LibreOffice Conference Milan, 2022

Understanding the Bug

Collabora Productivity

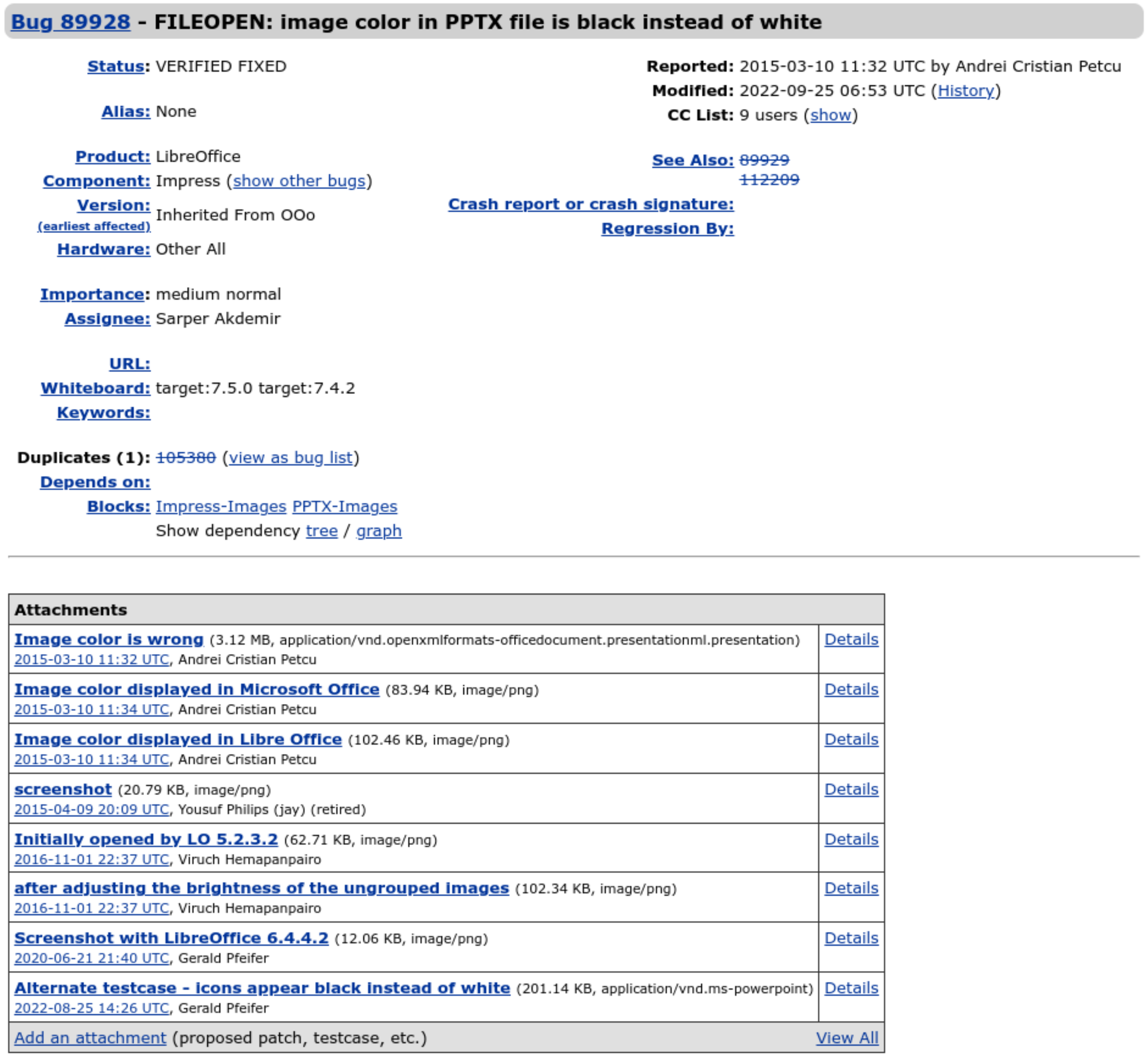
LibreOffice Conference Milan, 2022

Bug report

Investigate carefully

- Title (can be easily misleading…)
- Description
- Comments

Collabora Productivity



Bug 89928 - **FILEOPEN: image color in PPTX file is black instead of white**

Status: VERIFIED FIXED

Alias: None

Product: LibreOffice
Component: Impress (show other bugs)
Version: (earliest affected) Inherited From OOo
Hardware: Other All

Importance: medium normal
Assignee: Sarper Akdemir

URL:
Whiteboard: target:7.5.0 target:7.4.2
Keywords:

Duplicates (1): ~~105380~~ (view as bug list)
Depends on:
Blocks: Impress-Images PPTX-Images
Show dependency tree / graph

Reported: 2015-03-10 11:32 UTC by Andrei Cristian Petcu
Modified: 2022-09-25 06:53 UTC (History)
CC List: 9 users (show)

See Also: ~~89929~~ ~~112209~~
Crash report or crash signature:
Regression By:

Attachments	
Image color is wrong (3.12 MB, application/vnd.openxmlformats-officedocument.presentationml.presentation) 2015-03-10 11:32 UTC, Andrei Cristian Petcu	Details
Image color displayed in Microsoft Office (83.94 KB, image/png) 2015-03-10 11:34 UTC, Andrei Cristian Petcu	Details
Image color displayed in Libre Office (102.46 KB, image/png) 2015-03-10 11:34 UTC, Andrei Cristian Petcu	Details
screenshot (20.79 KB, image/png) 2015-04-09 20:09 UTC, Yousuf Philips (jay) (retired)	Details
Initially opened by LO 5.2.3.2 (62.71 KB, image/png) 2016-11-01 22:37 UTC, Viruch Hemapanpairo	Details
after adjusting the brightness of the ungrouped images (102.34 KB, image/png) 2016-11-01 22:37 UTC, Viruch Hemapanpairo	Details
Screenshot with LibreOffice 6.4.4.2 (12.06 KB, image/png) 2020-06-21 21:40 UTC, Gerald Pfeifer	Details
Alternate testcase - icons appear black instead of white (201.14 KB, application/vnd.ms-powerpoint) 2022-08-25 14:26 UTC, Gerald Pfeifer	Details
Add an attachment (proposed patch, testcase, etc.)	View All

LibreOffice Conference Milan, 2022

Investigating the Bug Documents

The most important resource!

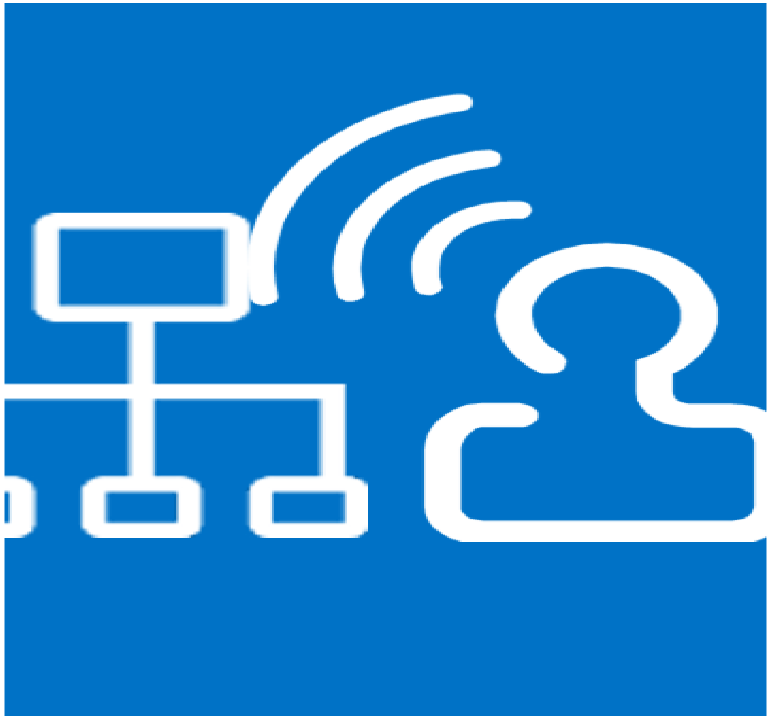
For import:

- Compare on Impress and PowerPoint
- Try to reproduce the problematic part in PowerPoint
- Explore the produced file (unzip & browse contents)

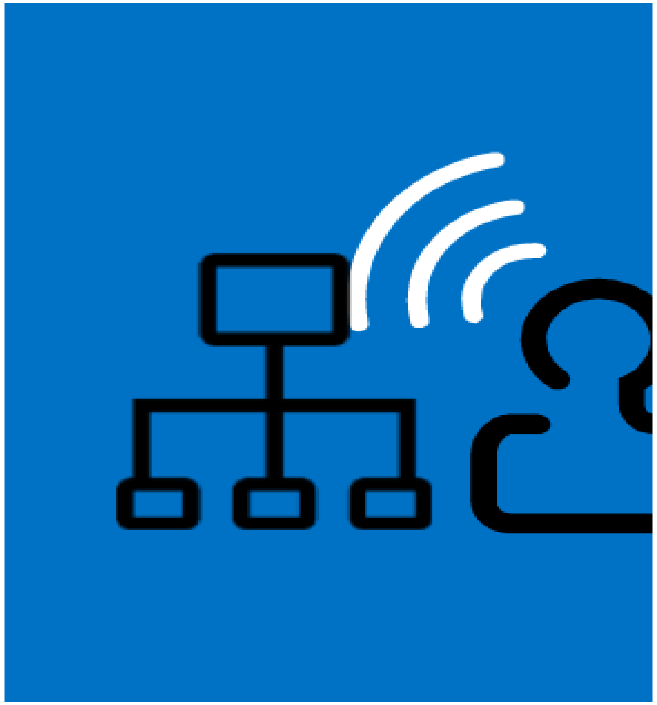
Collabora Productivity

LibreOffice Conference Milan, 2022

Investigating the Bug Documents



PowerPoint

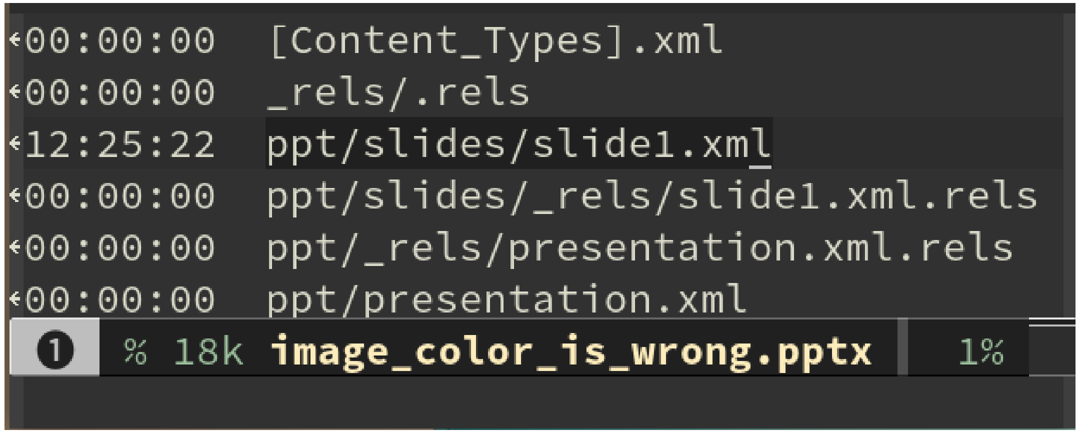


Impress

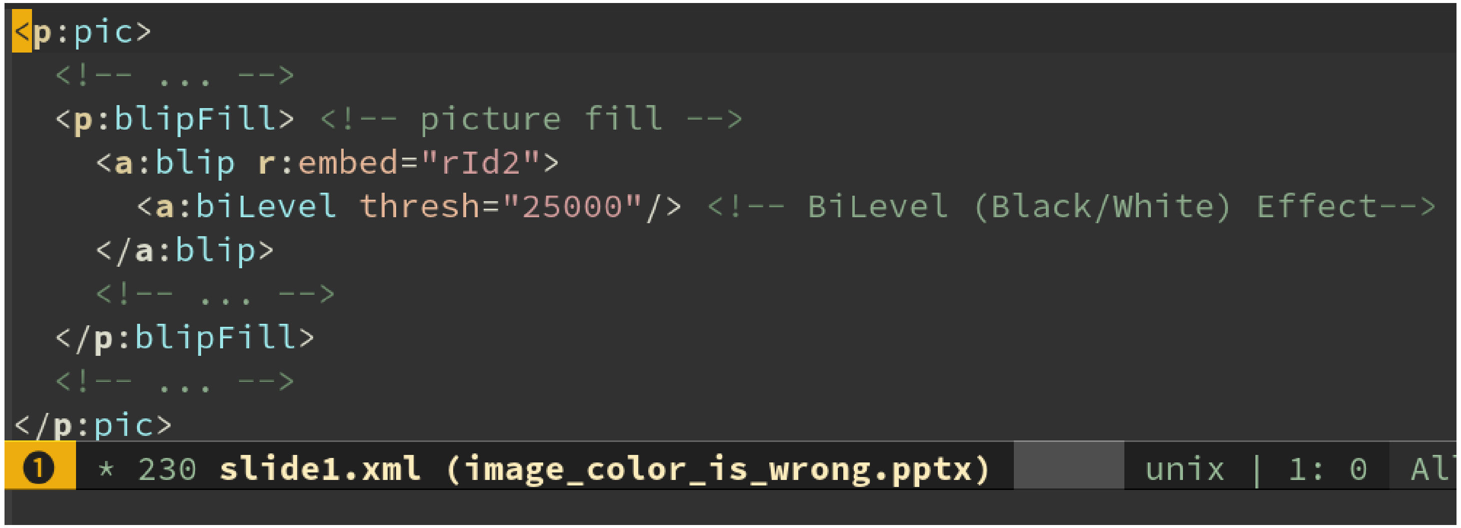
Isolate the problematic part!

Collabora Productivity

Investigating the Bug Documents



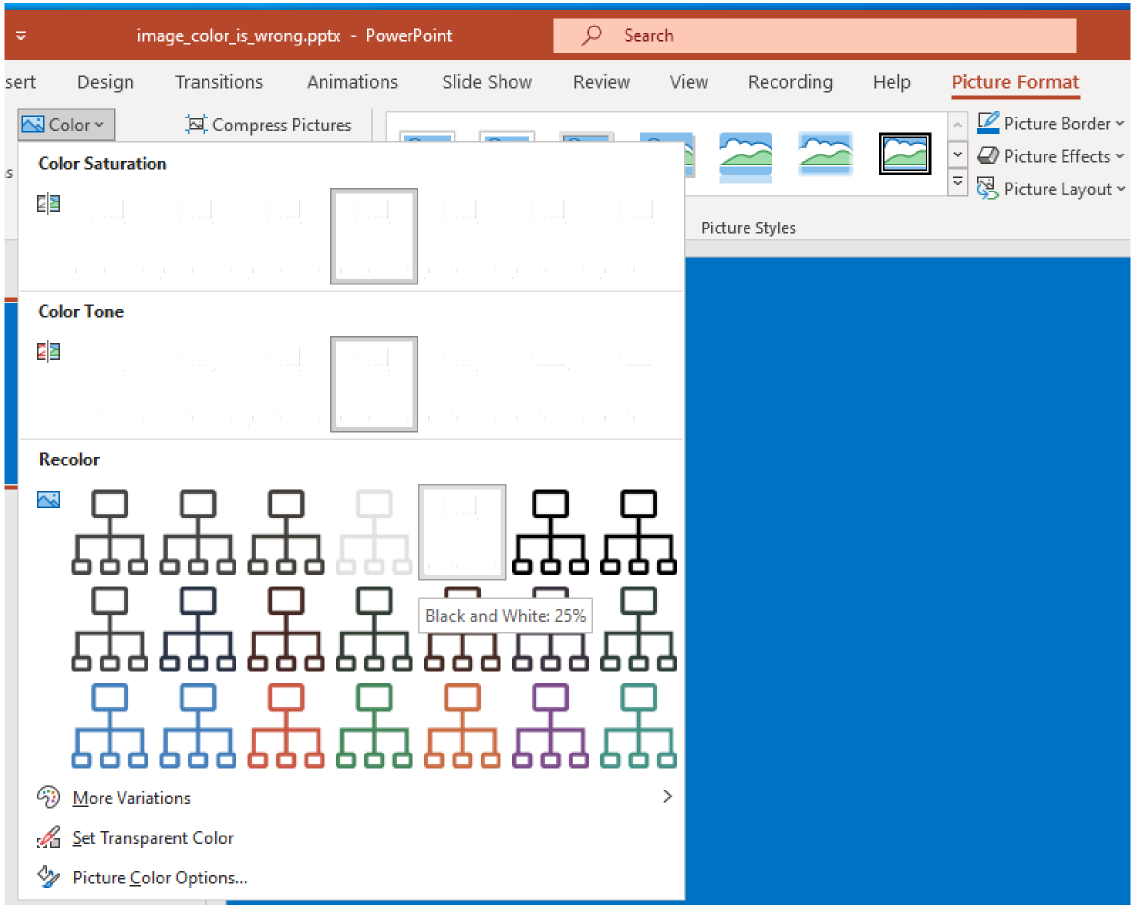
```
00:00:00  [Content_Types].xml
00:00:00  _rels/.rels
12:25:22  ppt/slides/slide1.xml
00:00:00  ppt/slides/_rels/slide1.xml.rels
00:00:00  ppt/_rels/presentation.xml.rels
00:00:00  ppt/presentation.xml
1  % 18k  image_color_is_wrong.pptx    1%
```



```
<p:pic>
  <!-- ... -->
  <p:blipFill> <!-- picture fill -->
    <a:blip r:embed="rId2">
      <a:biLevel thresh="25000"/> <!-- BiLevel (Black/White) Effect-->
    </a:blip>
    <!-- ... -->
  </p:blipFill>
  <!-- ... -->
</p:pic>
1  * 230  slide1.xml (image_color_is_wrong.pptx)    unix | 1: 0   Al
```

LibreOffice Conference Milan, 2022

Investigating the Bug Documents

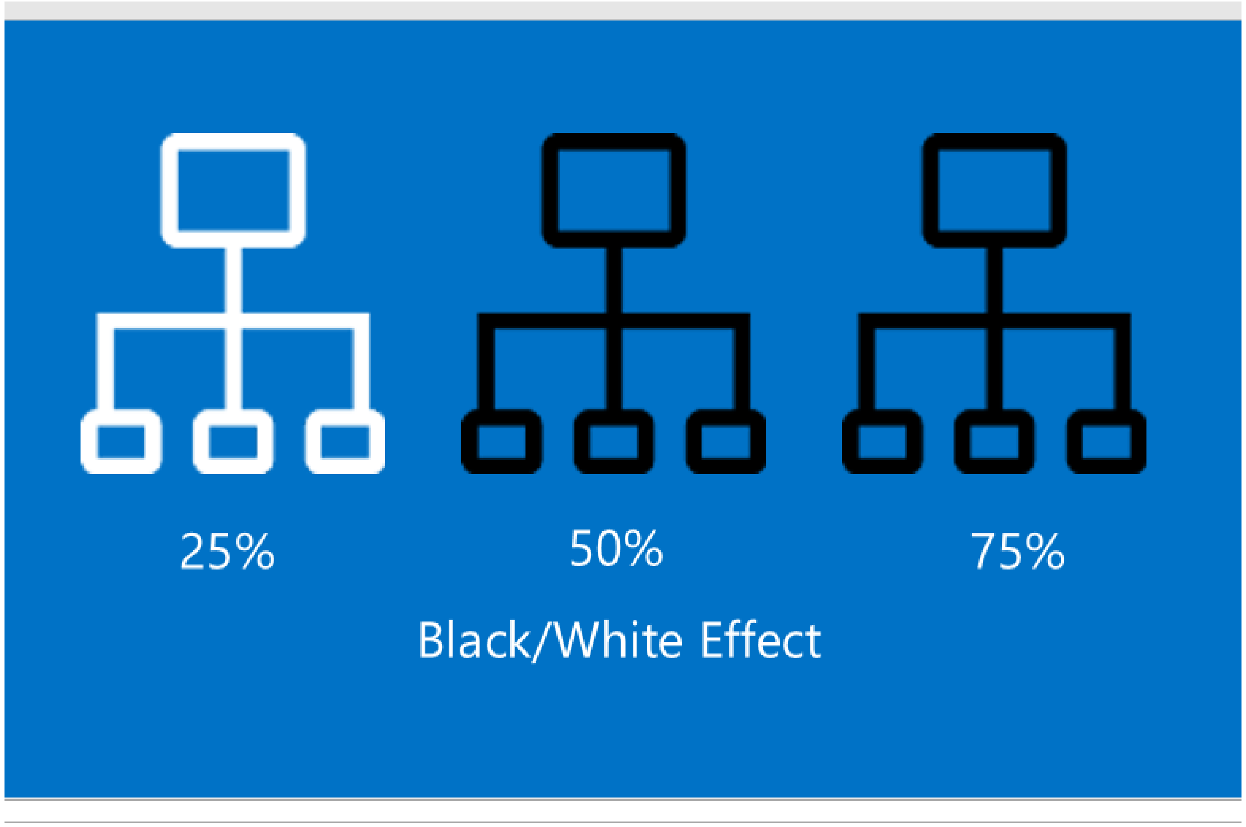


My hunch was "Impress doesn't have the Black/White Effect"

Collabora Productivity

LibreOffice Conference Milan, 2022

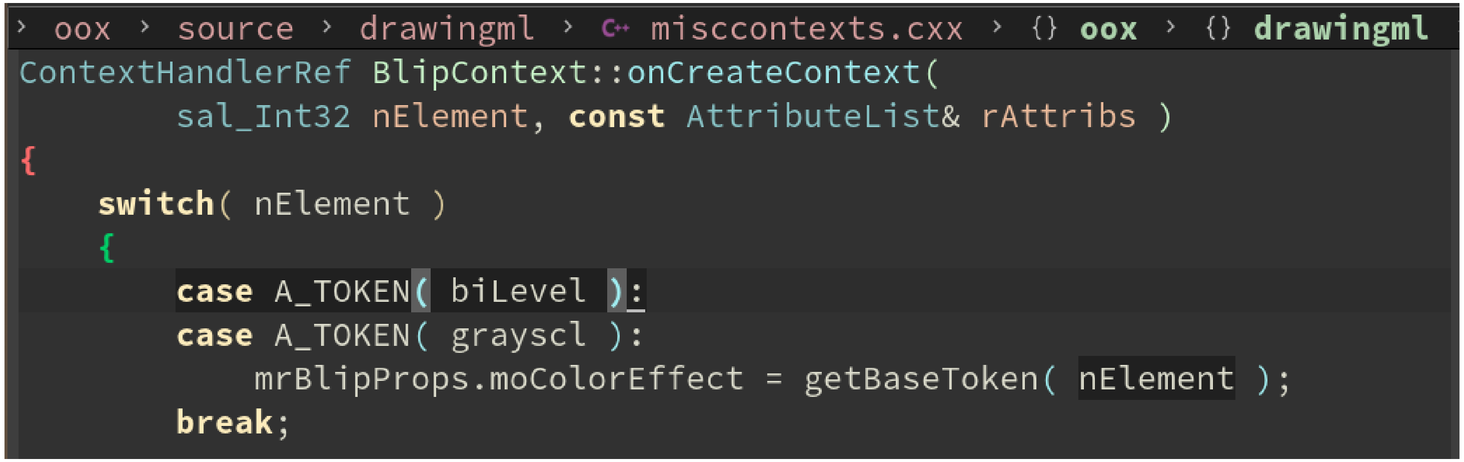
Investigating the Bug Documents



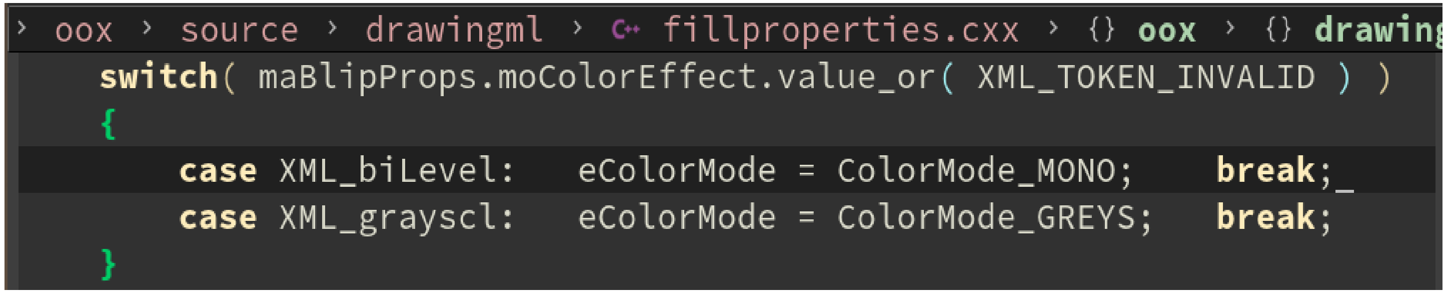
Collabora Productivity

LibreOffice Conference Milan, 2022

Investigating the Bug Documents



```
› oox › source › drawingml › misccontexts.cxx › {} oox › {} drawingml
ContextHandlerRef BlipContext::onCreateContext(
        sal_Int32 nElement, const AttributeList& rAttribs )
{
    switch( nElement )
    {
        case A_TOKEN( biLevel ):
        case A_TOKEN( grayscl ):
            mrBlipProps.moColorEffect = getBaseToken( nElement );
        break;
```



```
› oox › source › drawingml › fillproperties.cxx › {} oox › {} drawing
    switch( maBlipProps.moColorEffect.value_or( XML_TOKEN_INVALID ) )
    {
        case XML_biLevel:   eColorMode = ColorMode_MONO;    break;
        case XML_grayscl:   eColorMode = ColorMode_GREYS;   break;
    }
```

It looks like it biLevel gets resolved to ColorMode_MONO

Collabora Productivity

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Collabora Productivity

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Bug reason might be:

- No implemented import
- Non existent feature
- No straight forward way of mapping the feature

Collabora Productivity

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Initial solution draft. Will likely change during the implementation.

Ideal Solution would be:

- Generalized
    - Not just for the reported bug file
    - Introduces a missing concept in it’s totality
- Gives us the exact result visually with PowerPoint
- Doesn’t break existing working cases
- Easy to implement

Collabora Productivity

Coming up with a proposed fix

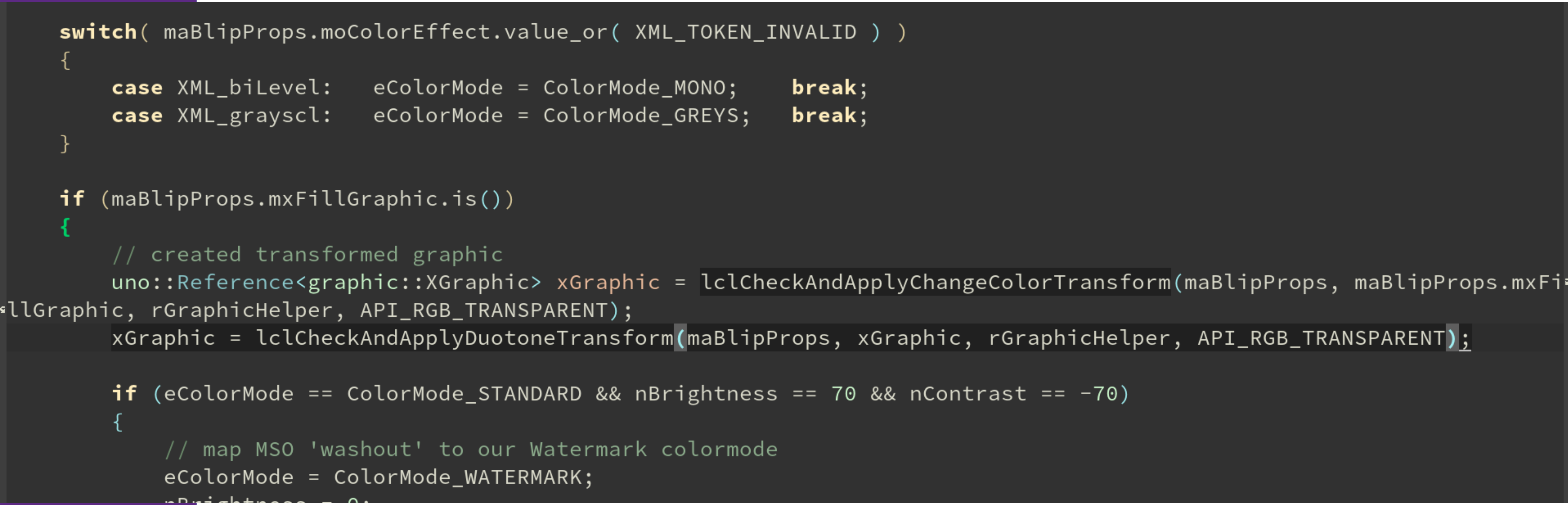
For the case with tdf#89928:

- Generalized:
  - Implement the missing parts of the feature
    - Color modes with thresholds 25%, 50%… (doc-model)
    - These values can be set through UI
    - Import & Export of these for odp
- Easy to implement
  - Apply the effect directly to the graphic during import

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Quick inspection:



```
    switch( maBlipProps.moColorEffect.value_or( XML_TOKEN_INVALID ) )
    {
        case XML_biLevel:   eColorMode = ColorMode_MONO;     break;
        case XML_grayscl:   eColorMode = ColorMode_GREYS;    break;
    }

    if (maBlipProps.mxFillGraphic.is())
    {
        // created transformed graphic
        uno::Reference<graphic::XGraphic> xGraphic = lclCheckAndApplyChangeColorTransform(maBlipProps, maBlipProps.mxFi
llGraphic, rGraphicHelper, API_RGB_TRANSPARENT);
        xGraphic = lclCheckAndApplyDuotoneTransform(maBlipProps, xGraphic, rGraphicHelper, API_RGB_TRANSPARENT);

        if (eColorMode == ColorMode_STANDARD && nBrightness == 70 && nContrast == -70)
        {
            // map MSO 'washout' to our Watermark colormode
            eColorMode = ColorMode_WATERMARK;
            nBrightness = 0;
```

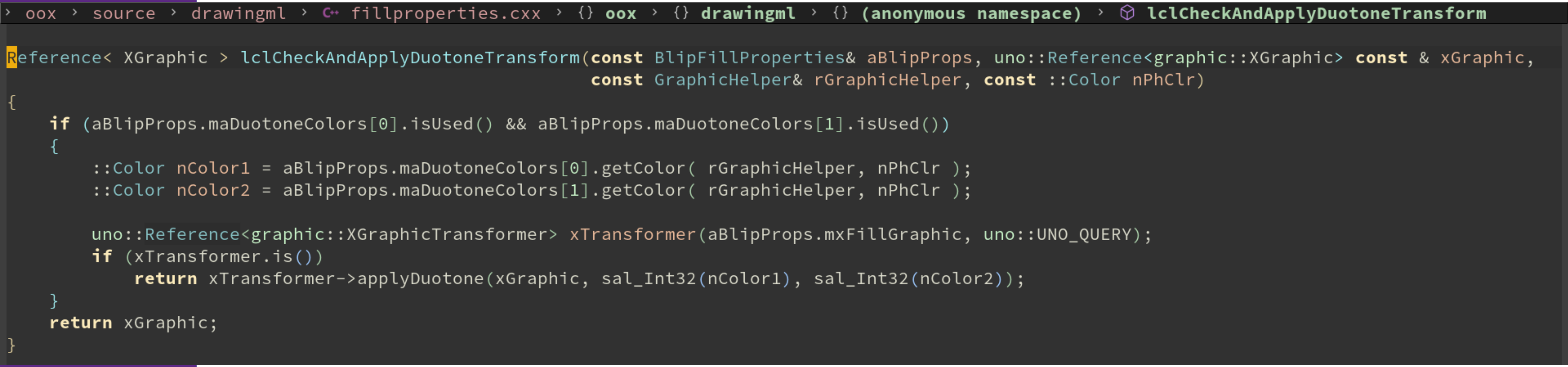
Easy to Implement → seems suitable

Collabora Productivity

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Quick inspection:



```
› oox › source › drawingml › fillproperties.cxx › {} oox › {} drawingml › {} (anonymous namespace) › lclCheckAndApplyDuotoneTransform

Reference< XGraphic > lclCheckAndApplyDuotoneTransform(const BlipFillProperties& aBlipProps, uno::Reference<graphic::XGraphic> const & xGraphic,
                                                       const GraphicHelper& rGraphicHelper, const ::Color nPhClr)
{
    if (aBlipProps.maDuotoneColors[0].isUsed() && aBlipProps.maDuotoneColors[1].isUsed())
    {
        ::Color nColor1 = aBlipProps.maDuotoneColors[0].getColor( rGraphicHelper, nPhClr );
        ::Color nColor2 = aBlipProps.maDuotoneColors[1].getColor( rGraphicHelper, nPhClr );

        uno::Reference<graphic::XGraphicTransformer> xTransformer(aBlipProps.mxFillGraphic, uno::UNO_QUERY);
        if (xTransformer.is())
            return xTransformer->applyDuotone(xGraphic, sal_Int32(nColor1), sal_Int32(nColor2));
    }
    return xGraphic;
}
```

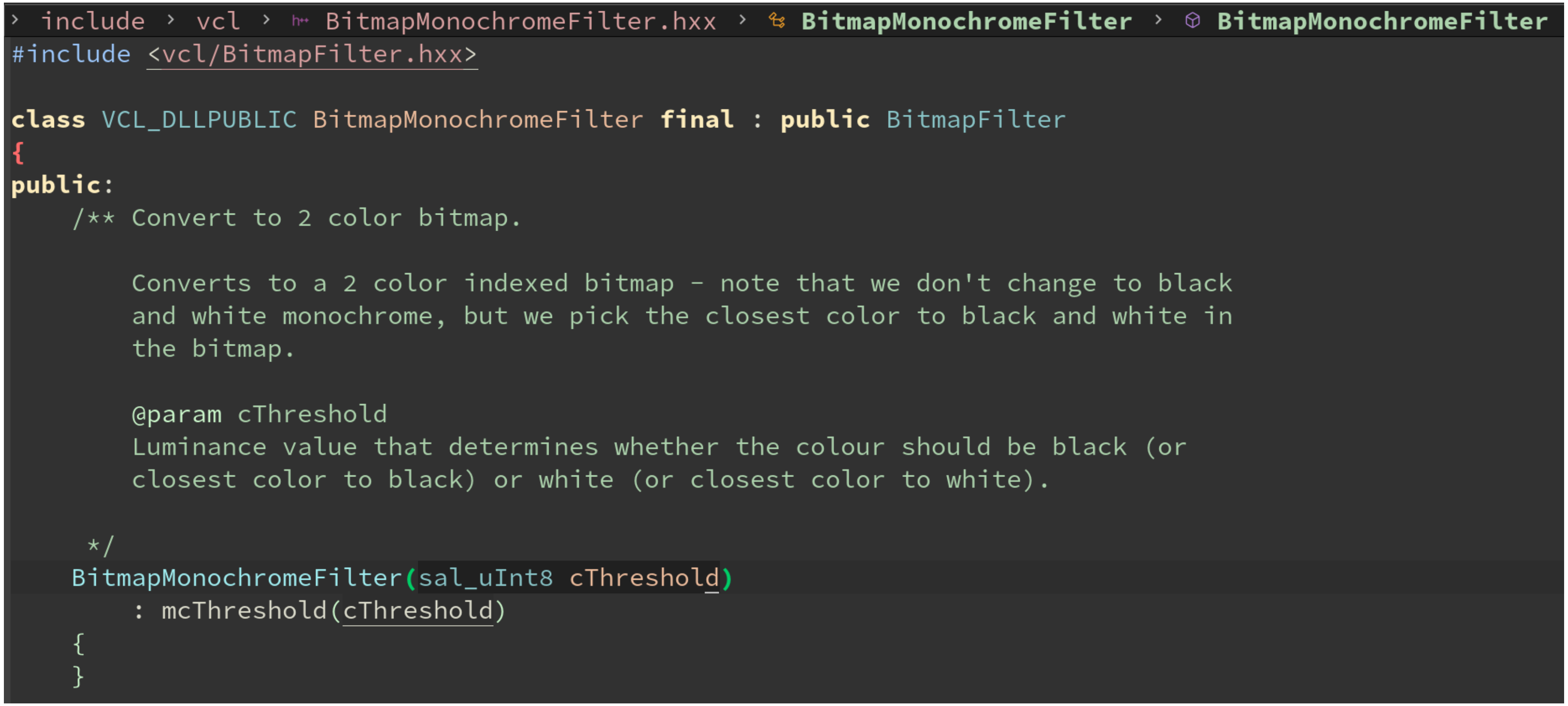
Easy to Implement → seems suitable

Collabora Productivity

LibreOffice Conference Milan, 2022

Coming up with a proposed fix

Quick inspection:



```
› include › vcl › BitmapMonochromeFilter.hxx › BitmapMonochromeFilter › BitmapMonochromeFilter
#include <vcl/BitmapFilter.hxx>

class VCL_DLLPUBLIC BitmapMonochromeFilter final : public BitmapFilter
{
public:
    /** Convert to 2 color bitmap.

        Converts to a 2 color indexed bitmap - note that we don't change to black
        and white monochrome, but we pick the closest color to black and white in
        the bitmap.

        @param cThreshold
        Luminance value that determines whether the colour should be black (or
        closest color to black) or white (or closest color to white).

     */
    BitmapMonochromeFilter(sal_uInt8 cThreshold)
        : mcThreshold(cThreshold)
    {
    }
```

Collabora Productivity

Coming up with a proposed fix

What we need to implement:

- Import of the threshold value
- Apply the Black/White effect considering this value (baked)

What we shouldn't break:

- Import of the already working ColorMode_MONO case!
  - Turns out this is the same with threshold value of 50%

Implementing the actual fix

LibreOffice Conference Milan, 2022

Implementing the actual fix

Where?

PPTX import stuff are in:

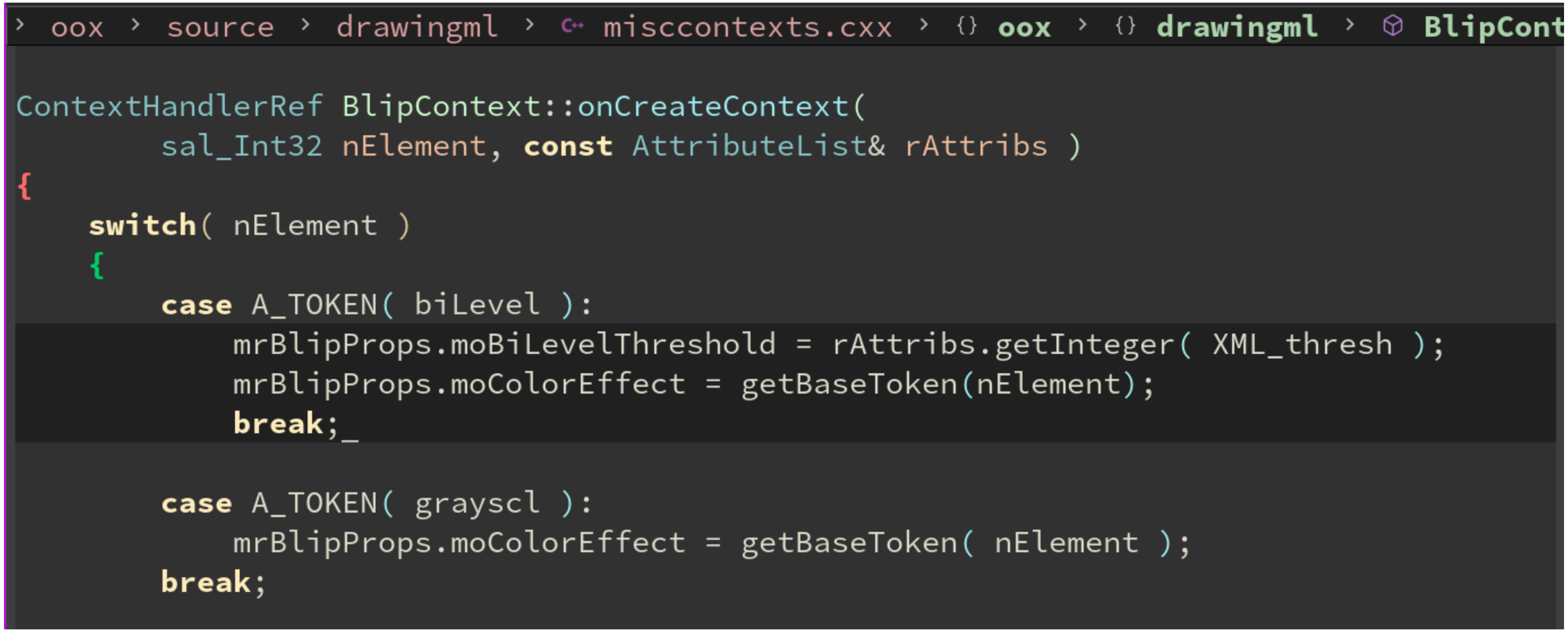
- oox/source/ppt
- oox/source/drawingml
- oox/source/*

Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing the actual fix

Import of the missing threshold value



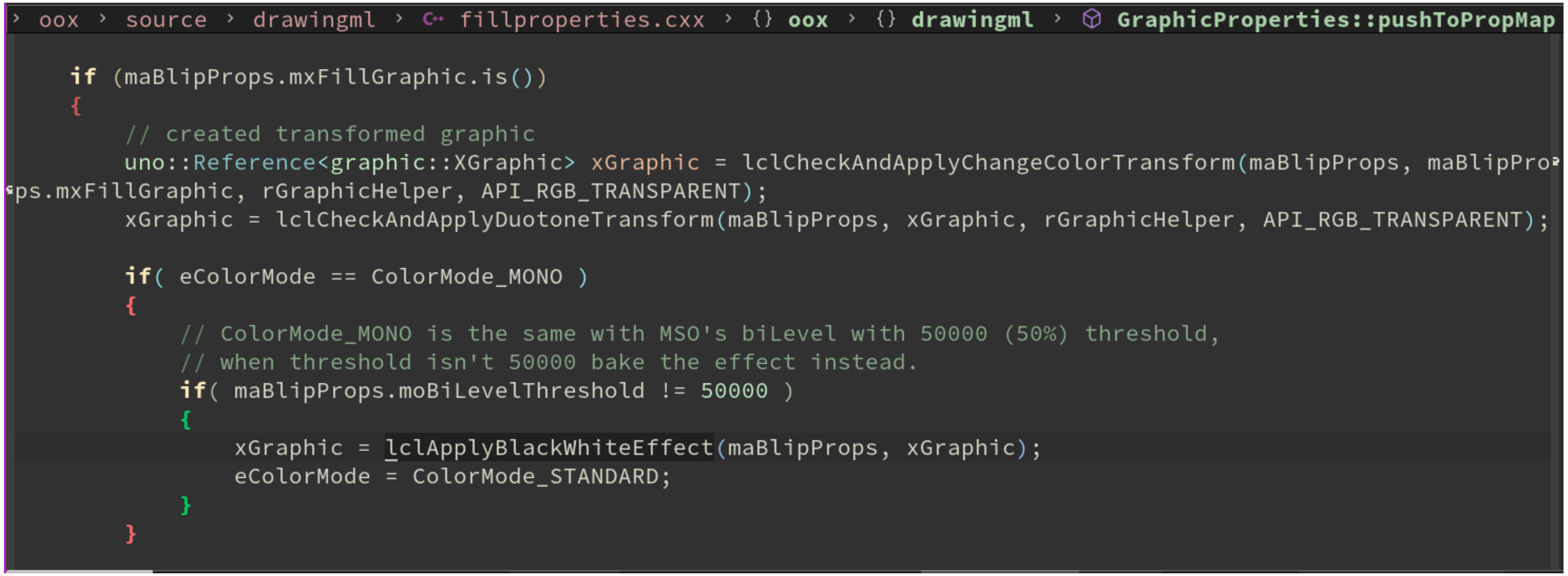
```
› oox › source › drawingml › misccontexts.cxx › {} oox › {} drawingml › BlipCont

ContextHandlerRef BlipContext::onCreateContext(
        sal_Int32 nElement, const AttributeList& rAttribs )
{
    switch( nElement )
    {
        case A_TOKEN( biLevel ):
            mrBlipProps.moBiLevelThreshold = rAttribs.getInteger( XML_thresh );
            mrBlipProps.moColorEffect = getBaseToken(nElement);
            break;_

        case A_TOKEN( grayscl ):
            mrBlipProps.moColorEffect = getBaseToken( nElement );
        break;
```

Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing the actual fix

Don’t break the case where ColorMode_MONO used to work!



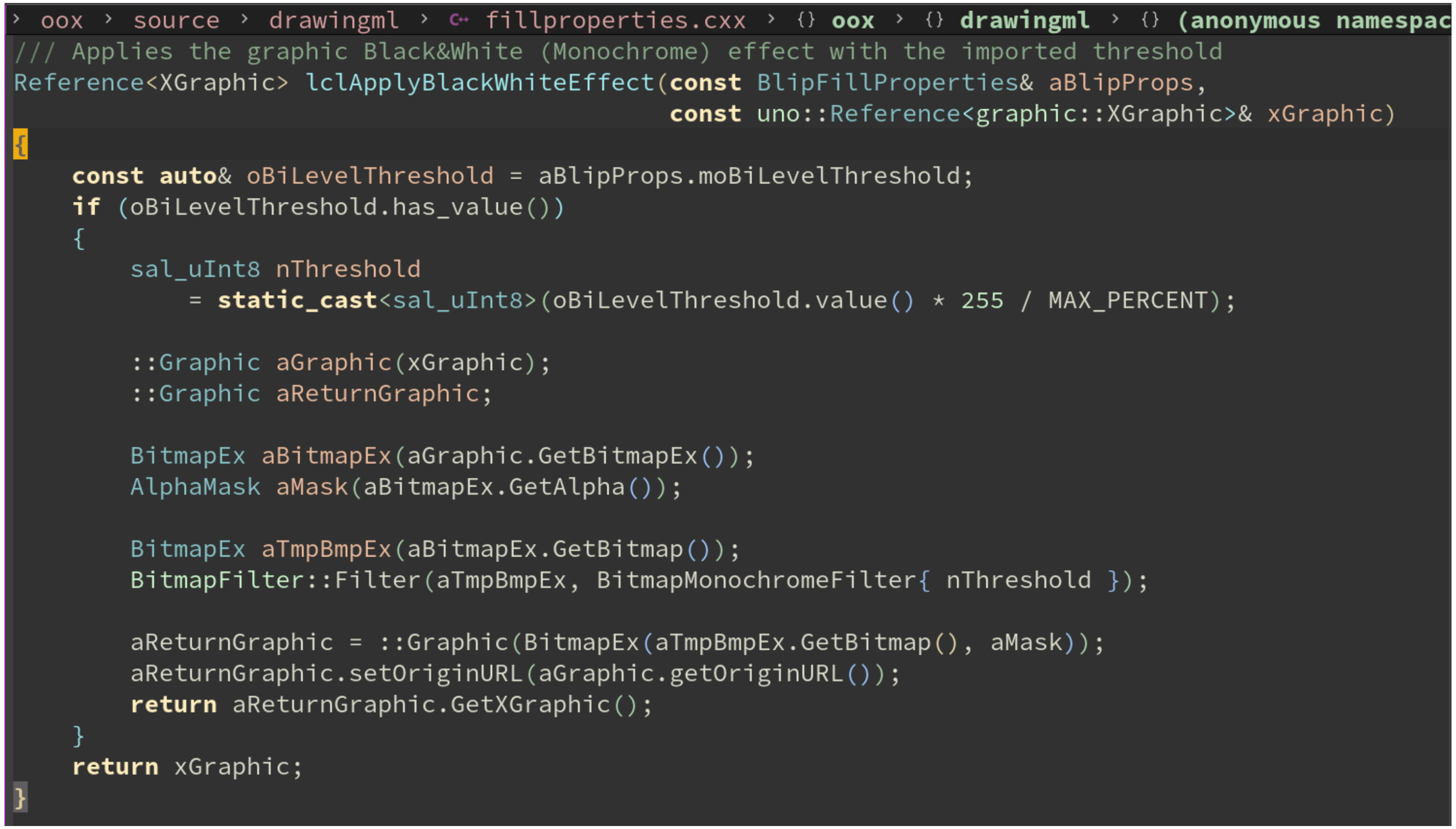
```
› oox › source › drawingml › fillproperties.cxx › {} oox › {} drawingml › GraphicProperties::pushToPropMap

    if (maBlipProps.mxFillGraphic.is())
    {
        // created transformed graphic
        uno::Reference<graphic::XGraphic> xGraphic = lclCheckAndApplyChangeColorTransform(maBlipProps, maBlipPro
ps.mxFillGraphic, rGraphicHelper, API_RGB_TRANSPARENT);
        xGraphic = lclCheckAndApplyDuotoneTransform(maBlipProps, xGraphic, rGraphicHelper, API_RGB_TRANSPARENT);

        if( eColorMode == ColorMode_MONO )
        {
            // ColorMode_MONO is the same with MSO's biLevel with 50000 (50%) threshold,
            // when threshold isn't 50000 bake the effect instead.
            if( maBlipProps.moBiLevelThreshold != 50000 )
            {
                xGraphic = lclApplyBlackWhiteEffect(maBlipProps, xGraphic);
                eColorMode = ColorMode_STANDARD;
            }
        }
```

Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing the actual fix

Apply the Black/White Effect considering threshold



› oox › source › drawingml › fillproperties.cxx › {} oox › {} drawingml › {} (anonymous namespac

```
/// Applies the graphic Black&White (Monochrome) effect with the imported threshold
Reference<XGraphic> lclApplyBlackWhiteEffect(const BlipFillProperties& aBlipProps,
                                             const uno::Reference<graphic::XGraphic>& xGraphic)
{
    const auto& oBiLevelThreshold = aBlipProps.moBiLevelThreshold;
    if (oBiLevelThreshold.has_value())
    {
        sal_uInt8 nThreshold
            = static_cast<sal_uInt8>(oBiLevelThreshold.value() * 255 / MAX_PERCENT);

        ::Graphic aGraphic(xGraphic);
        ::Graphic aReturnGraphic;

        BitmapEx aBitmapEx(aGraphic.GetBitmapEx());
        AlphaMask aMask(aBitmapEx.GetAlpha());

        BitmapEx aTmpBmpEx(aBitmapEx.GetBitmap());
        BitmapFilter::Filter(aTmpBmpEx, BitmapMonochromeFilter{ nThreshold });

        aReturnGraphic = ::Graphic(BitmapEx(aTmpBmpEx.GetBitmap(), aMask));
        aReturnGraphic.setOriginURL(aGraphic.getOriginURL());
        return aReturnGraphic.GetXGraphic();
    }
    return xGraphic;
}
```

Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing tests for the fix

Collabora Productivity

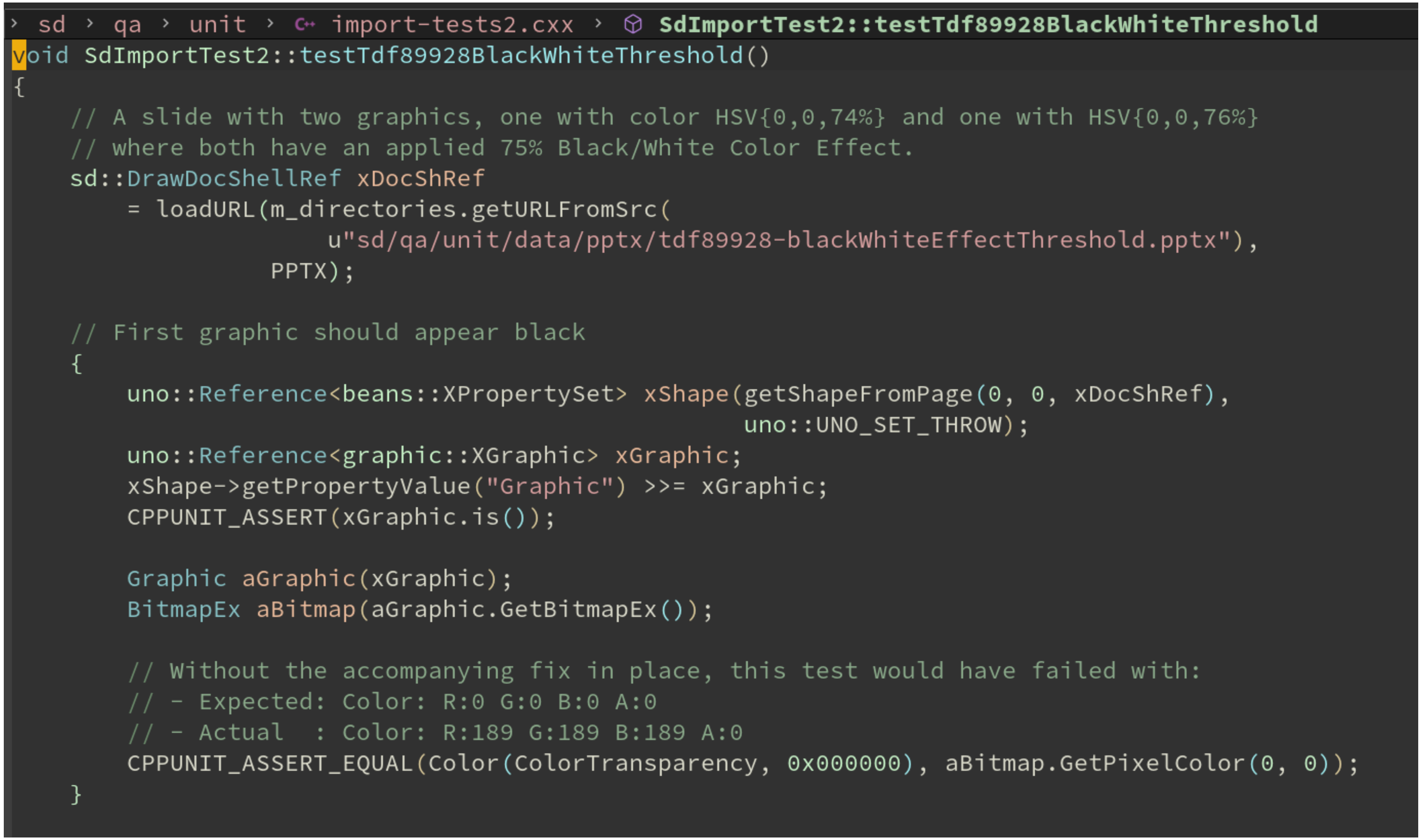
Implementing tests for the fix

What you can write your test for

- What you’ve just fixed
- What was already working
- What you think might accidentally break!

LibreOffice Conference Milan, 2022

Implementing tests for the fix



```
› sd › qa › unit › import-tests2.cxx › SdImportTest2::testTdf89928BlackWhiteThreshold
void SdImportTest2::testTdf89928BlackWhiteThreshold()
{
    // A slide with two graphics, one with color HSV{0,0,74%} and one with HSV{0,0,76%}
    // where both have an applied 75% Black/White Color Effect.
    sd::DrawDocShellRef xDocShRef
        = loadURL(m_directories.getURLFromSrc(
                      u"sd/qa/unit/data/pptx/tdf89928-blackWhiteEffectThreshold.pptx"),
                  PPTX);

    // First graphic should appear black
    {
        uno::Reference<beans::XPropertySet> xShape(getShapeFromPage(0, 0, xDocShRef),
                                                   uno::UNO_SET_THROW);
        uno::Reference<graphic::XGraphic> xGraphic;
        xShape->getPropertyValue("Graphic") >>= xGraphic;
        CPPUNIT_ASSERT(xGraphic.is());

        Graphic aGraphic(xGraphic);
        BitmapEx aBitmap(aGraphic.GetBitmapEx());

        // Without the accompanying fix in place, this test would have failed with:
        // - Expected: Color: R:0 G:0 B:0 A:0
        // - Actual  : Color: R:189 G:189 B:189 A:0
        CPPUNIT_ASSERT_EQUAL(Color(ColorTransparency, 0x000000), aBitmap.GetPixelColor(0, 0));
    }
```

Collabora Productivity

Export Bug (PPTX)

Understanding the Bug

LibreOffice Conference Milan, 2022

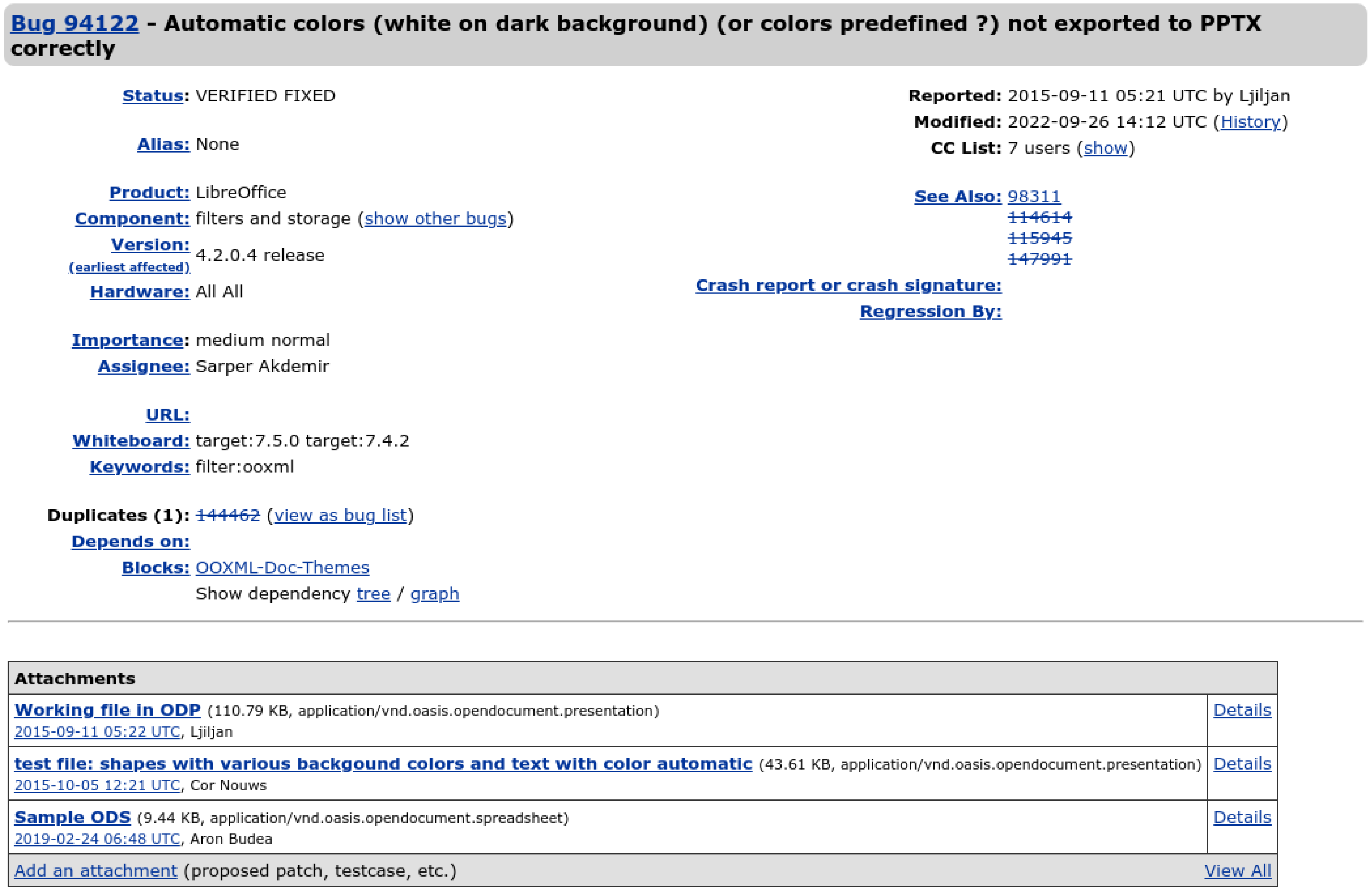
Bug report

Investigate carefully

- Title (can be easily misleading…)
- Description
- Comments

Collabora Productivity

LibreOffice Conference Milan, 2022



Bug 94122 - Automatic colors (white on dark background) (or colors predefined ?) not exported to PPTX correctly

Status:	VERIFIED FIXED
Alias:	None
Product:	LibreOffice
Component:	filters and storage (show other bugs)
Version: (earliest affected)	4.2.0.4 release
Hardware:	All All
Importance:	medium normal
Assignee:	Sarper Akdemir
URL:	
Whiteboard:	target:7.5.0 target:7.4.2
Keywords:	filter:ooxml
Duplicates (1):	~~144462~~ (view as bug list)
Depends on:	
Blocks:	OOXML-Doc-Themes
	Show dependency tree / graph

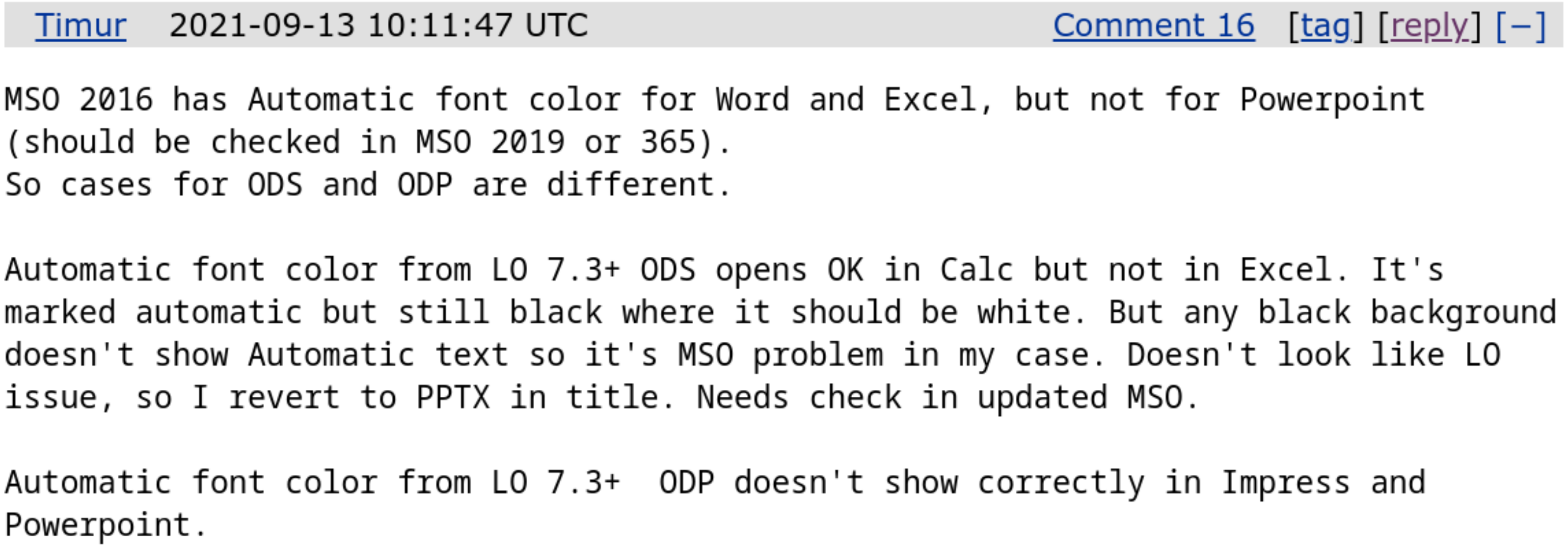
Reported:	2015-09-11 05:21 UTC by Ljiljan
Modified:	2022-09-26 14:12 UTC (History)
CC List:	7 users (show)
See Also:	98311 ~~114614~~ ~~115945~~ ~~147991~~
Crash report or crash signature:	
Regression By:	

Attachments	
Working file in ODP (110.79 KB, application/vnd.oasis.opendocument.presentation) 2015-09-11 05:22 UTC, Ljiljan	Details
test file: shapes with various backgound colors and text with color automatic (43.61 KB, application/vnd.oasis.opendocument.presentation) 2015-10-05 12:21 UTC, Cor Nouws	Details
Sample ODS (9.44 KB, application/vnd.oasis.opendocument.spreadsheet) 2019-02-24 06:48 UTC, Aron Budea	Details
Add an attachment (proposed patch, testcase, etc.)	View All

Collabora Productivity

LibreOffice Conference Milan, 2022

Bug report



Timur 2021-09-13 10:11:47 UTC Comment 16 [tag] [reply] [−]

MSO 2016 has Automatic font color for Word and Excel, but not for Powerpoint (should be checked in MSO 2019 or 365).
So cases for ODS and ODP are different.

Automatic font color from LO 7.3+ ODS opens OK in Calc but not in Excel. It's marked automatic but still black where it should be white. But any black background doesn't show Automatic text so it's MSO problem in my case. Doesn't look like LO issue, so I revert to PPTX in title. Needs check in updated MSO.

Automatic font color from LO 7.3+ ODP doesn't show correctly in Impress and Powerpoint.

Collabora Productivity

Bug report

- Word & Excel has automatic colors
- PowerPoint doesn’t!

Coming up with a proposed fix

Coming up with a proposed fix

Initial solution draft. Will likely change during the implementation.

Ideal Solution would be:

- Generalized
  - Not just for the reported bug file
  - Introduces a missing concept in it’s totality
- Gives us the exact result visually with PowerPoint
- Doesn’t break existing working cases
- Easy to implement

Coming up with a proposed fix

For the case with tdf#94122:

- Observations:
  - COL_AUTO is White or Black whether the background is Dark or Light
  - COL_AUTO only cares about slide background & shape fill.
  - Importance: Shape Fill first, then Slide background

Automatic Color (no fill)

Automatic Color (dark fill)

Automatic Color (light fill)

Coming up with a proposed fix

For the case with tdf#94122:

- Generalized:
  - Resolve COL_AUTO just as Impress does natively
  - Export the resulting color
- Easy to implement
  - Resolve COL_AUTO by checking shape fill & slide background color.
  - Export the resulting color

LibreOffice Conference Milan, 2022

Implementing the fix

Collabora Productivity

Implementing the fix

Where?

PPTX export stuff are in:

- sd/source/filter/eppt (mostly in pptx-* files)
- oox/source/export

LibreOffice Conference Milan, 2022

Implementing the fix

Let’s try to implement the generalized case!
(Can we resolve the color just as Impress does natively?)

Grepping some code, thought these might work:
ImpEditEngine::GetAutoColor()
vcl::Font::GetColor() → (comment states it is pretty much obselete..)

Couldn’t get it to work…

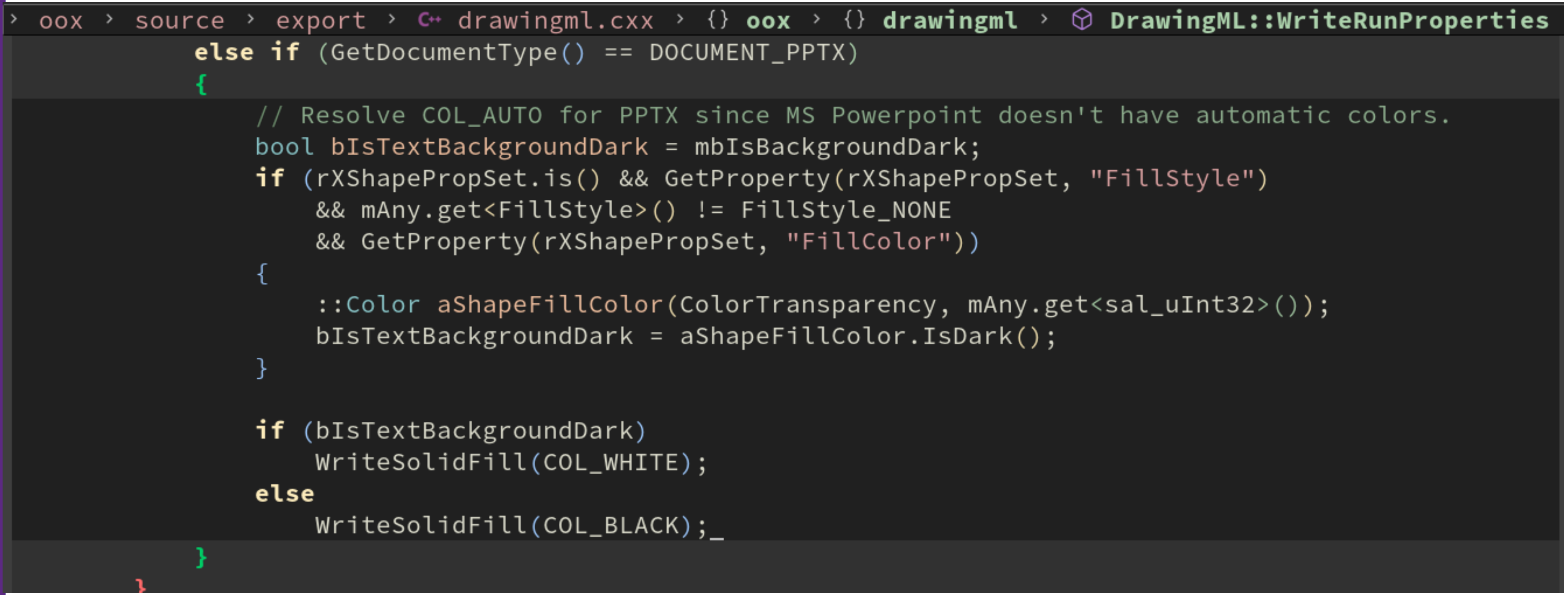
Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing the fix

Go with the “resolve color by checking the known conditions” way.

Easy enough!



› oox › source › export › drawingml.cxx › {} oox › {} drawingml › DrawingML::WriteRunProperties

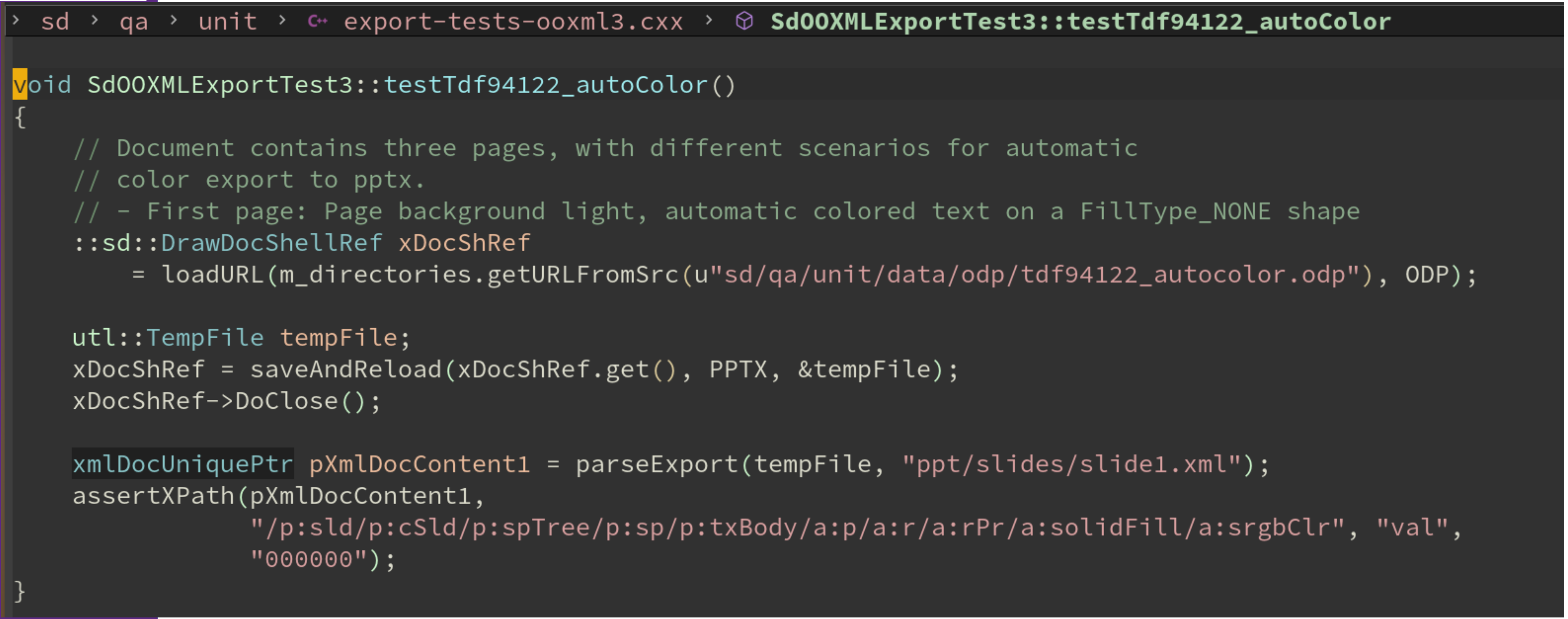
```
        else if (GetDocumentType() == DOCUMENT_PPTX)
        {
            // Resolve COL_AUTO for PPTX since MS Powerpoint doesn't have automatic colors.
            bool bIsTextBackgroundDark = mbIsBackgroundDark;
            if (rXShapePropSet.is() && GetProperty(rXShapePropSet, "FillStyle")
                && mAny.get<FillStyle>() != FillStyle_NONE
                && GetProperty(rXShapePropSet, "FillColor"))
            {
                ::Color aShapeFillColor(ColorTransparency, mAny.get<sal_uInt32>());
                bIsTextBackgroundDark = aShapeFillColor.IsDark();
            }

            if (bIsTextBackgroundDark)
                WriteSolidFill(COL_WHITE);
            else
                WriteSolidFill(COL_BLACK);_
        }
```

Collabora Productivity

Implementing tests for the fix

LibreOffice Conference Milan, 2022

Implementing tests for the fix



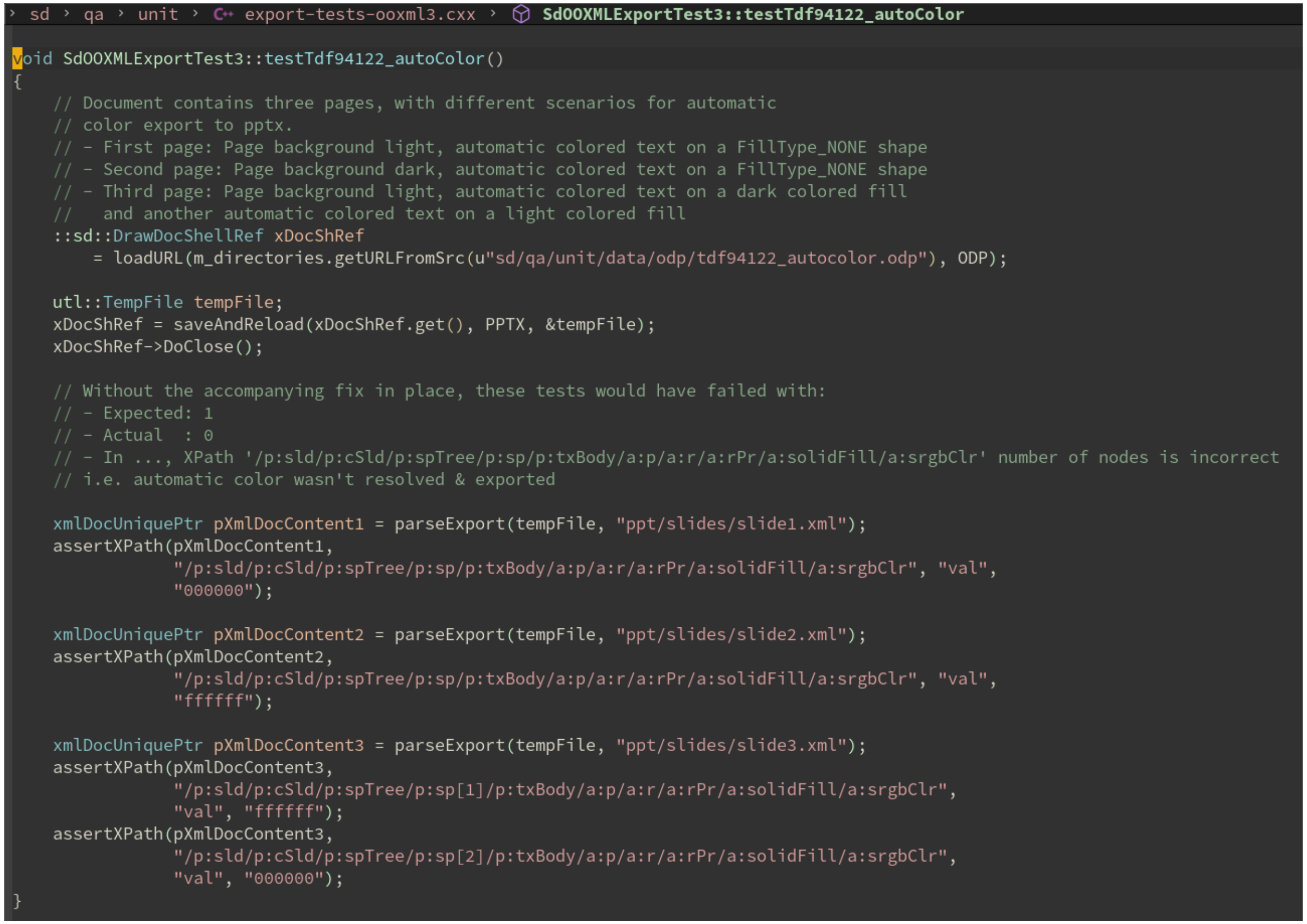
› sd › qa › unit › export-tests-ooxml3.cxx › SdOOXMLExportTest3::testTdf94122_autoColor

```
void SdOOXMLExportTest3::testTdf94122_autoColor()
{
    // Document contains three pages, with different scenarios for automatic
    // color export to pptx.
    // - First page: Page background light, automatic colored text on a FillType_NONE shape
    ::sd::DrawDocShellRef xDocShRef
        = loadURL(m_directories.getURLFromSrc(u"sd/qa/unit/data/odp/tdf94122_autocolor.odp"), ODP);

    utl::TempFile tempFile;
    xDocShRef = saveAndReload(xDocShRef.get(), PPTX, &tempFile);
    xDocShRef->DoClose();

    xmlDocUniquePtr pXmlDocContent1 = parseExport(tempFile, "ppt/slides/slide1.xml");
    assertXPath(pXmlDocContent1,
                "/p:sld/p:cSld/p:spTree/p:sp/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr", "val",
                "000000");
}
```

Collabora Productivity

LibreOffice Conference Milan, 2022

Implementing tests for the fix



```
› sd › qa › unit › export-tests-ooxml3.cxx › SdOOXMLExportTest3::testTdf94122_autoColor

void SdOOXMLExportTest3::testTdf94122_autoColor()
{
    // Document contains three pages, with different scenarios for automatic
    // color export to pptx.
    // - First page: Page background light, automatic colored text on a FillType_NONE shape
    // - Second page: Page background dark, automatic colored text on a FillType_NONE shape
    // - Third page: Page background light, automatic colored text on a dark colored fill
    //   and another automatic colored text on a light colored fill
    ::sd::DrawDocShellRef xDocShRef
        = loadURL(m_directories.getURLFromSrc(u"sd/qa/unit/data/odp/tdf94122_autocolor.odp"), ODP);

    utl::TempFile tempFile;
    xDocShRef = saveAndReload(xDocShRef.get(), PPTX, &tempFile);
    xDocShRef->DoClose();

    // Without the accompanying fix in place, these tests would have failed with:
    // - Expected: 1
    // - Actual  : 0
    // - In ..., XPath '/p:sld/p:cSld/p:spTree/p:sp/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr' number of nodes is incorrect
    // i.e. automatic color wasn't resolved & exported

    xmlDocUniquePtr pXmlDocContent1 = parseExport(tempFile, "ppt/slides/slide1.xml");
    assertXPath(pXmlDocContent1,
                "/p:sld/p:cSld/p:spTree/p:sp/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr", "val",
                "000000");

    xmlDocUniquePtr pXmlDocContent2 = parseExport(tempFile, "ppt/slides/slide2.xml");
    assertXPath(pXmlDocContent2,
                "/p:sld/p:cSld/p:spTree/p:sp/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr", "val",
                "ffffff");

    xmlDocUniquePtr pXmlDocContent3 = parseExport(tempFile, "ppt/slides/slide3.xml");
    assertXPath(pXmlDocContent3,
                "/p:sld/p:cSld/p:spTree/p:sp[1]/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr",
                "val", "ffffff");
    assertXPath(pXmlDocContent3,
                "/p:sld/p:cSld/p:spTree/p:sp[2]/p:txBody/a:p/a:r/a:rPr/a:solidFill/a:srgbClr",
                "val", "000000");
}
```

Collabora Productivity

LibreOffice Conference Milan, 2022

Thanks !

By Sarper Akdemir

Collabora Productivity

@CollaboraOffice
hello@collaboraoffice.com
Collaboraoffice.com