Chasing an Interoperability Bug in Impress

Sarper Akdemir

Consultant Software Engineer Intern

sarper.akdemir@collabora.com
@quwex
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
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
Import Bug (PPTX)
Understanding the Bug
Bug report

Investigate carefully

- Title (can be easily misleading...)
- Description
- Comments
Bug 89928 - FILEOPEN: image color in PPTX file is black instead of white

**Status:** VERIFIED FIXED

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

**Product:** LibreOffice
**Component:** Impress ([show other bugs](#))

**Version:** 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):** #65349 ([view as bug list](#))
**Depends on:**

**Blocks:** Impress-Images PPTX-Images

Show dependency tree / graph

### Attachments

<table>
<thead>
<tr>
<th>Attachment</th>
<th>Size</th>
<th>Description</th>
<th>Date</th>
<th>Username</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image color is wrong</strong> (3.12 MB, application/vnd.openxmlformats-officedocument.presentationml.presentation)</td>
<td>2015-03-10 11:32 UTC</td>
<td>Andrei Cristian Petcu</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>Image color displayed in Microsoft Office</strong> (83.94 KB, image/png)</td>
<td>2015-03-10 11:34 UTC</td>
<td>Andrei Cristian Petcu</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>Image color displayed in LibreOffice</strong> (102.46 KB, image/png)</td>
<td>2015-03-10 11:34 UTC</td>
<td>Andrei Cristian Petcu</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>screenshot</strong> (20.79 KB, image/png)</td>
<td>2015-04-09 20:95 UTC</td>
<td>Yusuf Philips (joy) (retired)</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>Initially opened by LO 5.2.3.2</strong> (92.71 KB, image/png)</td>
<td>2016-11-01 01:22:37 UTC</td>
<td>Vinod Hemanaprag</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>after adjusting the brightness of the ungroupped images</strong> (102.34 KB, image/png)</td>
<td>2016-11-01 01:22:37 UTC</td>
<td>Vinod Hemanaprag</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>Screenshot with LibreOffice 6.4.4.2</strong> (12.06 KB, image/png)</td>
<td>2020-06-21 21:40 UTC</td>
<td>Gerald Millier</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td><strong>Alternate testcase - Icons appear black instead of white</strong> (201.14 KB, application/vnd.ms-powerpoint)</td>
<td>2022-08-15 14:26 UTC</td>
<td>Gerald Millier</td>
<td></td>
<td>Details</td>
<td></td>
</tr>
</tbody>
</table>

**Add an attachment** (proposed patch, testcase, etc.) [View All](#)
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)
Investigating the Bug Documents

PowerPoint

Isolate the problematic part!

Impress
Investigating the Bug Documents

```xml
<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>
```
Investigating the Bug Documents

My hunch was “Impress doesn’t have the Black/White Effect”
Investigating the Bug Documents

Black/White Effect

25%  50%  75%
Investigating the Bug Documents

```
ContextHandlerRef BlipContext::onCreateContext(
    sal_Int32 nElement, const AttributeList& rAttrs)
{
    switch( nElement )
    {
    case A_TOKEN( biLevel):
    case A_TOKEN( grayscl ):
        mrBlipProps.moColorEffect = getToken( nElement );
        break;
    }
```

```
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
Coming up with a proposed fix
Coming up with a proposed fix

Bug reason might be:

- No implemented import
- Non existent feature
- No straight forward way of mapping the feature
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#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
Coming up with a proposed fix

Quick inspection:

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

if (maBlipProps.mxFillGraphic.is())
{
    // created transformed graphic
    uno::Reference< graphic::XGraphic > xGraphic = lclCheckAndApplyChangeColorTransform(maBlipProps, maBlipProps.mxFillGraphic, 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
Coming up with a proposed fix

Quick inspection:

Easy to Implement → seems suitable
Coming up with a proposed fix

Quick inspection:

```cpp
#include <vcl/vcl.hxx>
#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)
    {
    }
```
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
Implementing the actual fix

Where?

PPTX import stuff are in:

- oox/source/ppt
- oox/source/drawingml
- oox/source/*
Implementing the actual fix

Import of the missing threshold value

```cpp
ContextHandlerRef BlipContext::onCreateContext(
    sal_Int32 nElement, const AttributeList& rAttrs)
{
    switch (nElement)
    {
    case A_TOKEN( biLevel):
        mrBlipProps.moBiLevelThreshold = rAttrs.getInteger( XML_thres );
        mrBlipProps.moColorEffect = getToken(nElement);
        break;

    case A_TOKEN( grayscl ) :
        mrBlipProps.moColorEffect = getToken(nElement);
        break;
    ```
Implementing the actual fix

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

```cpp
if (maBlipProps.mxFillGraphic.is())
{
    // created transformed graphic
    uno::Reference<graph::XGraphic> xGraphic = lclCheckAndApplyChangeColorTransform(maBlipProps, maBlipProps.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;
        }
    }
```
Implementing the actual fix

Apply the Black/White Effect considering threshold

```cpp
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;
```
Implementing tests for the fix
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!
Implementing tests for the fix

```cpp
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));
    }
```
Export Bug (PPTX)
Understanding the Bug
Bug report

Investigate carefully

- Title (can be easily misleading...)
- Description
- Comments
Bug 94122 - Automatic colors (white on dark background) (or colors predefined ?) not exported to PPTX correctly

**Status:** VERIFIED FIXED

**Assigned:** None

**Product:** LibreOffice

**Component:** filters and storage (show other bugs)

**Version:** 4.2.0.4 release

**Hardware:** All All

**Importance:** medium normal

**Assignee:** Serper Akdemir

**URL:**

**Whiteboard:** target:7.5.0 target:7.4.2

**Keywords:** filter:ooxml

**Duplicates:** 114462 (view as bug list)

**Depends on:**

- Blocks: OOXML-Doc-Themes

Show dependency tree / graph

---

**Attachments**

- **Working file in ODP** (110.79 KB, application/vnd.oasis.opendocument.presentation)
  2015-09-11 05:22 UTC, Ljiljan

- **test file: shapes with various background colors and text with color automatic** (43.61 KB, application/vnd.oasis.opendocument.presentation)
  2015-10-05 12:21 UTC, Cor Nouws

- **Sample ODS** (9.44 KB, application/vnd.oasis.opendocument.spreadsheet)
  2019-02-24 06:48 UTC, Aron Buica

Add an attachment (proposed patch, testcase, etc.)
Bug report

Timur 2021-09-13 10:11:47 UTC

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.
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
Implementing the fix
Implementing the fix

Where?

PPTX export stuff are in:

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

Couldn’t get it to work...
Implementing the fix

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

```cpp
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);
}
```
Implementing tests for the fix
Implementing tests for the fix

```cpp
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,
               "000000");
}
```
Implementing tests for the fix

```cpp
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
    ::std::DrawDocShellRef xDocShellRef
        = loadURL(m_directories.getURLFromSrc(u"sd/qa/unit/data/odp/tdf94122_autocolor.odp"), ODP);

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

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

    xmlDocUniquePtr pXmlDocContent1 = parseExport(tempFile, "p:slides/slide1.xml");
    assertXPath(pXmlDocContent1,

    xmlDocUniquePtr pXmlDocContent2 = parseExport(tempFile, "p:slides/slide2.xml");
    assertXPath(pXmlDocContent2,

    xmlDocUniquePtr pXmlDocContent3 = parseExport(tempFile, "p:slides/slide3.xml");
    assertXPath(pXmlDocContent3,
    assertXPath(pXmlDocContent3,
}
```
Thanks!

By Sarper Akdemir