OOXML Document Analysis

By Gülşah Köse

Software Engineer Collabora Online

@gulsahkse

How do we proceed when we receive a problematic OOXML document?

Reproduce the problem

There may be more than one problem in same document. We must make sure we produce the right one.

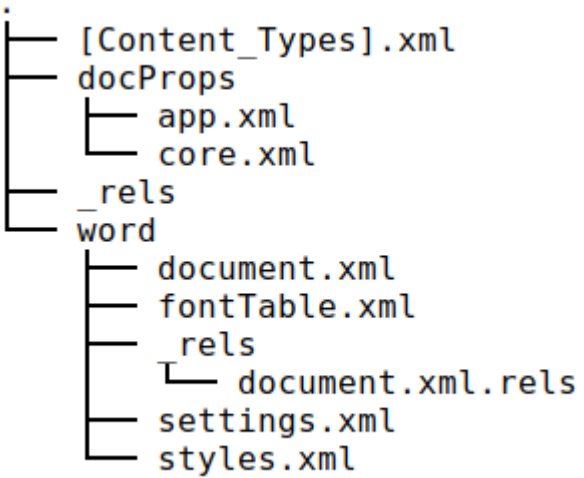
Simplify the document

The document may contain texts, images, sounds. But the problem might just about the images. If possible, other elements in the document can be cleaned.

Collabora Online

Read the XML

OOXML documents (docx, pptx, xlsx etc) contains compressed XML files. Result of the “unzip test.docx” command will be seen like this.



Detect the XML part problematic element

For example: A paragraph element will look like as following

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="Normal"/>
    <w:bidi w:val="0"/>
    <w:jc w:val="left"/>
    <w:rPr/>
  </w:pPr>
  <w:r>
    <w:rPr/>
    <w:t>Hello</w:t>
  </w:r>
</w:p>
```

Collabora Online

Read the OOXML specs

It can be understood at this stage whether the syntax is appropriate or not. Apart from that, it allows us to understand the element in more detail than what I see in the interface.

Pick the keywords

We will need this to understand how Collabora Office handles these elements.

grep, read, debug loop

It cannot be formulated after this stage.

Reading code → detecting suspicious parts → debug them is a loop.

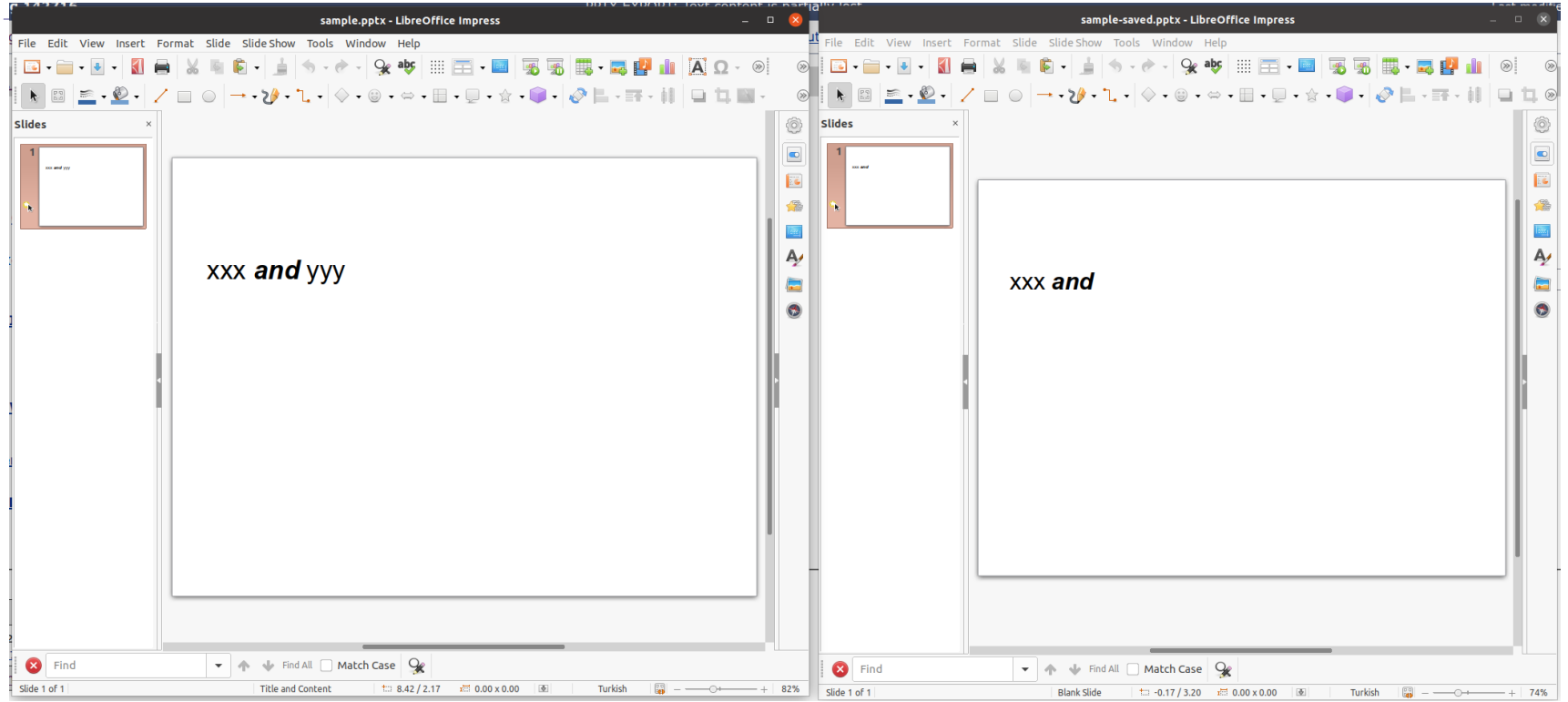
Often writing code is a small part of that proces :)

Real example

Bug report: https://bugs.documentfoundation.org/show_bug.cgi?id=142716
(PPTX EXPORT: Text content is partially lost)

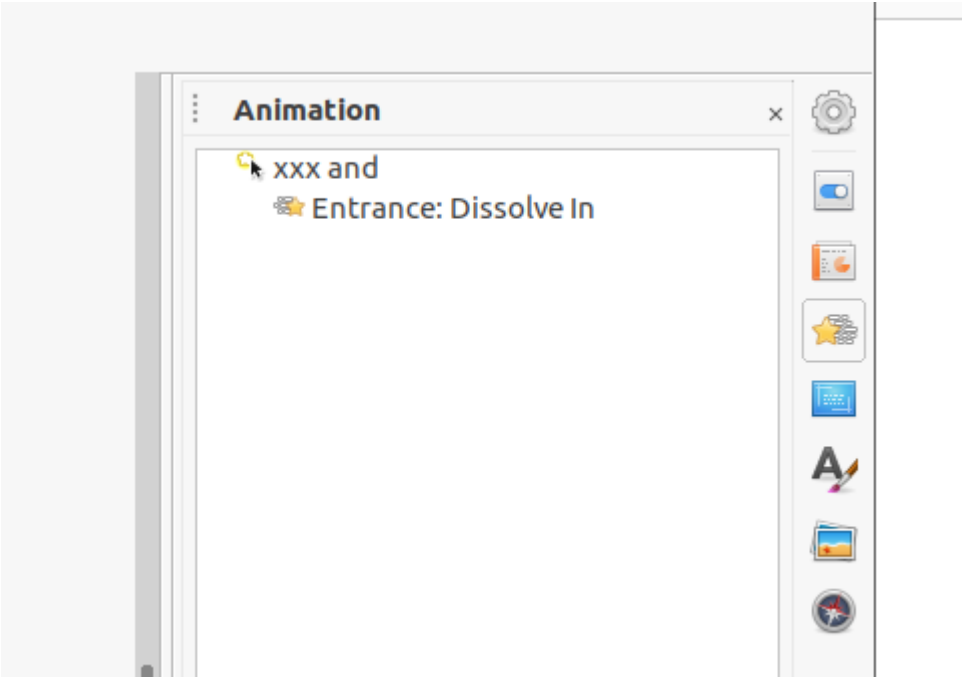
Before

After



```
<p:txBody>
  <a:bodyPr/>
  <a:lstStyle/>
  <a:p>
    <a:r>
      <a:rPr lang="tr-TR" dirty="0" smtClean="0"/>
      <a:t>xxx</a:t>
    </a:r>
    <a:r>
      <a:rPr lang="en-NZ" dirty="0" smtClean="0"/>
      <a:t> </a:t>
    </a:r>
    <a:r>
      <a:rPr lang="en-NZ" b="1" i="1" dirty="0" smtClean="0"/>
      <a:t>and </a:t>
    </a:r>
    <a:r>
      <a:rPr lang="tr-TR" dirty="0" smtClean="0"/>
      <a:t>yyy</a:t>
    <a:endParaRPr lang="en-US" dirty="0"/>
  </a:p>
</p:txBody>
```

(text/date via Insert>Header and Footer)


Animation
xxx and
Entrance: Dissolve In

```
<p:animEffect transition="in" filter="dissolve">
  <p:cBhvr>
    <p:cTn id="7" dur="500"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:pRg st="0" end="0"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

/media/gulsah/EK/libreoffice-public-2/editeng/source/uno/unotext.cxx

```
1875
1876        uno::Reference< text::XTextRange > SAL_CALL SvxUnoTextBase::getEnd()
1877        {
1878            return SvxUnoTextRangeBase::getEnd();
1879        }
1880
1881        OUString SAL_CALL SvxUnoTextBase::getString()
1882        {
1883            return SvxUnoTextRangeBase::getString();
1884        }
1885
1886        void SAL_CALL SvxUnoTextBase::setString( const OUString& aString )
1887        {
1888            SvxUnoTextRangeBase::setString(aString);
1889        }
1890
1891
1892        // XEnumerationAccess
1893        uno::Reference< container::XEnumeration > SAL_CALL SvxUnoTextBase::createEnumeration()
B+>1894        {
1895            SolarMutexGuard aGuard;
1896            if( maSelection == ESelection(0,0,0,0) || maSelection == ESelection(EE_PARA_MAX_COUNT,0,0,0) )
1897            {
1898                ESelection aSelection;
1899                ::GetSelection( aSelection, GetEditSource()->GetTextForwarder() );
1900                return new SvxUnoTextContentEnumeration(*this, aSelection);
1901            }
1902            else
1903            {
1904                return new SvxUnoTextContentEnumeration(*this, maSelection);
1905            }
1906        }
1907
1908        // XElementAccess ( container::XEnumerationAccess )
1909        uno::Type SAL_CALL SvxUnoTextBase::getElementType(  )
1910        {
1911            return cppu::UnoType<text::XTextRange>::get();
1912        }
1913
1914        sal_Bool SAL_CALL SvxUnoTextBase::hasElements(  )
1915        {
```

multi-thre Thread 0x7fffe90d50 In: SvxUnoTextBase::createEnumeration L1894 PC: 0x7ffff6c212f0

```
Thread 1 "soffice.bin" hit Breakpoint 1, animcore::(anonymous namespace)::AnimationNode::createEnumeration (this=0x55555a0a4e70) at /media/gulsah/EK/libreoffice-public-2/animations/source/animcore/animcore.cxx:1906
(gdb) n
(gdb) c
Continuing.

Thread 1 "soffice.bin" hit Breakpoint 1, animcore::(anonymous namespace)::AnimationNode::createEnumeration (this=0x555556f647b0) at /media/gulsah/EK/libreoffice-public-2/animations/source/animcore/animcore.cxx:1906
Continuing.

Thread 1 "soffice.bin" hit Breakpoint 1, animcore::(anonymous namespace)::AnimationNode::createEnumeration (this=0x55555711f270) at /media/gulsah/EK/libreoffice-public-2/animations/source/animcore/animcore.cxx:1906
(gdb) c
Continuing.

Thread 1 "soffice.bin" hit Breakpoint 1, animcore::(anonymous namespace)::AnimationNode::createEnumeration (this=0x55555a20ed10) at /media/gulsah/EK/libreoffice-public-2/animations/source/animcore/animcore.cxx:1906
Continuing.

Thread 1 "soffice.bin" hit Breakpoint 1, non-virtual thunk to SvxUnoTextBase::createEnumeration() () at /media/gulsah/EK/libreoffice-public-2/include/editeng/unotext.hxx:451
(gdb) c
Continuing.

Thread 1 "soffice.bin" hit Breakpoint 1, SvxUnoTextBase::createEnumeration (this=0x555559e75ab8) at /media/gulsah/EK/libreoffice-public-2/editeng/source/uno/unotext.cxx:1894
(gdb)
```

main | attrlist.cxx | mem-cpu | script | srv | Shell

libreoffice-public-2 : gdb - Drop-Down Terminal

Solution: https://gerrit.libreoffice.org/c/core/+/117557

```
void SAL_CALL SvxUnoTextBase::insertString( const uno::Reference< text::XTextRange >& xRange, const
OUString& aString, sal_Bool bAbsorb )
{
    SolarMutexGuard aGuard;

    if( !xRange.is() )
        return;

    if (GetEditSource())
    {
        ESelection aSelection;
        ::GetSelection( aSelection, GetEditSource()->GetTextForwarder() );
        SetSelection( aSelection );
    }

    SvxUnoTextRangeBase* pRange = comphelper::getUnoTunnelImplementation<SvxUnoTextRange>( xRange );
    if(!pRange)
        return;

    // setString on SvxUnoTextRangeBase instead of itself QuickInsertText
    // and UpdateData, so that the selection will be adjusted to
    // SvxUnoTextRangeBase. Actually all cursor objects of this Text must
    // to be statement to be adapted!

    if (!bAbsorb)                   // do not replace -> append on tail
        pRange->CollapseToEnd();

    pRange->setString( aString );

    pRange->CollapseToEnd();
}
```

```
void SAL_CALL SvxUnoTextBase::insertString( const uno::Reference< text::XTextRange >& xRange, const
OUString& aString, sal_Bool bAbsorb )
{
    SolarMutexGuard aGuard;

    if( !xRange.is() )
        return;

    SvxUnoTextRangeBase* pRange = comphelper::getUnoTunnelImplementation<SvxUnoTextRange>( xRange );
    if(!pRange)
        return;

    // setString on SvxUnoTextRangeBase instead of itself QuickInsertText
    // and UpdateData, so that the selection will be adjusted to
    // SvxUnoTextRangeBase. Actually all cursor objects of this Text must
    // to be statement to be adapted!

    if (!bAbsorb)                   // do not replace -> append on tail
        pRange->CollapseToEnd();

    pRange->setString( aString );

    pRange->CollapseToEnd();

    if (GetEditSource())
    {
        ESelection aSelection;
        ::GetSelection( aSelection, GetEditSource()->GetTextForwarder() );
        SetSelection( aSelection );
    }
}
```

(text/date via Insert>Header and Footer)

Thanks !

By Gülşah Köse

@CollaboraOffice
hello@collaboraoffice.com
Collaboraoffice.com