

Cleaning up the unittest code mess

Xisco Fauli

`xiscofauli@libreoffice.org`

Jenkins failing on Windows

```
[ RUN _____] VBAMacroTest::testMultiDocumentCopyAndPaste
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element xsi:type http://www.w3.org/2001/XMLSchema-instance
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element xsi:type http://www.w3.org/2001/XMLSchema-instance
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1248: unknown attribute revIDLastSave=0
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1248: unknown attribute documentId=13_ncr:1_{F7A034E9-7B1C-4BBF-A4D3-1621E552C012}
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element xr6:coauthVersionLast http://schemas.microsoft.com/office/spreadsheetml/2016/revision6
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element xr6:coauthVersionMax http://schemas.microsoft.com/office/spreadsheetml/2016/revision6
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element xr10:uidLastSave http://schemas.microsoft.com/office/spreadsheetml/2016/revision10
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1248: unknown attribute vid={4A3C46E8-61CC-4603-A589-7422A47A8E4A}
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1324: unknown element x14ac:knownFonts http://schemas.microsoft.com/office/spreadsheetml/2009/9/ac
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1248: unknown attribute defaultSlicerStyle=SlicerStyleLight1
warn:xmloff:21030:19428892:sax/source/fastparser/fastparser.cxx:1248: unknown attribute defaultTimelineStyle=TimeSlicerStyleLight1
warn:xmloff:21030:19429253:sax/source/fastparser/fastparser.cxx:1324: unknown element xr:uid http://schemas.microsoft.com/office/spreadsheetml/2014/revision
```

Error: a unit test failed, please do one of:

```
make CppunitTest_sc_vba_macro_test CPPUNITTRACE="lldb --" # for interactive debugging on macOS
make CppunitTest_sc_vba_macro_test VALGRIND=memcheck # for memory checking
```

You can limit the execution to just one particular test by:

```
make CppunitTest_sc_vba_macro_test CPPUNIT_TEST_NAME="testXYZ" ...above mentioned params...
```

```
/Users/tdf/lode/jenkins/workspace/lo_gerrit/Config/macosx_clang_dbgutil/solenv/gbuild/CppunitTest.mk:119: recipe for target '/Users/tdf/lode/jenkins/workspace/lo_gerrit/Config/macosx_clang_dbgutil/workdir/CppunitTest/sc_vba_macro_test.test' failed
make[1]: *** [/Users/tdf/lode/jenkins/workspace/lo_gerrit/Config/macosx_clang_dbgutil/workdir/CppunitTest/sc_vba_macro_test.test] Error 1
Makefile:289: recipe for target 'build' failed
make: *** [build] Error 2
Build was aborted
Finished: FAILURE
```

PROBLEM: tearDown method disposing mxComponent was missing.
fixed thanks to Noel Grandin

Main tests classes

CppUnit::TestFixture



BootstrapFixture

- validate
- IsDefaultDPI
- getDPIScaling

MacroTest

- loadFromDesktop
- dispatchCommand
- parseExportStream

XmlTestTools

- assertXPath
- parseXmlStream

FiltersTest

- testDir
- load/save

Goal

CppUnit::TestFixture



BootstrapFixture

- validate
- IsDefaultDPI
- getDPIScaling

MacroTest

- loadFromDesktop
- dispatchCommand
- parseExportStream

XmlTestTools

- assertXPath
- parseXmlStream

FiltersTest

- testDir
- load/save

UnoApiTest

UnoApiXmlTest

Clean ups

- Factor out setUp/tearDown methods
 - Found some test classes missed the tearDown method
- Define mxComponent in UnoApiTest directly
 - Found some tests were using their own xComponent instead of the class member
- Merge CalcUnoApiTest into UnoApiTest
 - Found some test classes in sc inherited from CalcUnoApiTest which inherited from UnoApiTest for no real reason
- Other methods factored out
 - ParsePDFExport, executeMacro, createTempCopy

Load/Save

- Sd and sc modules had each their own methods to load/save the files instead of using loadFromDesktop from MacroTest
 - Copy-pasted from FiltersTest
 - ~1.500 lines of code removed

```
-// These values are taken from "Flags" in filter/source/config/fragments/filters/*
-// You need to turn value of oor:name="Flags" to SfxFilterFlags::*, see
-// include/comphelper/documentconstants.hxx for the possible values.
-// Note: 3RDPARTYFILTER == SfxFilterFlags::STARONEFILTER
-#define ODP_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::TEMPLATE | SfxFilterFlags::OWN | SfxFilterFlags::DEFAULT | SfxFilterFlags::ENCRYPTION | SfxFilterFlags::PREFERRED )
-#define PPT_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::ALIEN )
-#define PPTX_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::ALIEN | SfxFilterFlags::STARONEFILTER | SfxFilterFlags::PREFERRED )
-#define HTML_FORMAT_TYPE ( SfxFilterFlags::EXPORT | SfxFilterFlags::ALIEN )
-#define PDF_FORMAT_TYPE ( SfxFilterFlags::STARONEFILTER | SfxFilterFlags::ALIEN | SfxFilterFlags::IMPORT | SfxFilterFlags::PREFERRED )
-#define FODG_FORMAT_TYPE ( SfxFilterFlags::STARONEFILTER | SfxFilterFlags::OWN | SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT )
-#define FODP_FORMAT_TYPE ( SfxFilterFlags::STARONEFILTER | SfxFilterFlags::OWN | SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT )
-#define SXI_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::TEMPLATE | SfxFilterFlags::OWN | SfxFilterFlags::ALIEN | SfxFilterFlags::PREFERRED | SfxFilterFlags::ENCRYPTION )
-#define ODG_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::TEMPLATE | SfxFilterFlags::OWN | SfxFilterFlags::DEFAULT | SfxFilterFlags::ENCRYPTION | SfxFilterFlags::PREFERRED )
-#define PPTM_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::ALIEN | SfxFilterFlags::STARONEFILTER | SfxFilterFlags::PREFERRED )
-#define POTX_FORMAT_TYPE ( SfxFilterFlags::IMPORT | SfxFilterFlags::EXPORT | SfxFilterFlags::ALIEN | SfxFilterFlags::TEMPLATE | SfxFilterFlags::STARONEFILTER | SfxFilterFlags::PREFERRED )
```

```

-Taken from filter/source/config/fragments/filters/ too:
-pName: The file extension.
-pFilterName: <node oor:Name="...">
-pTypeName: <prop oor:Name="Type">...</prop>
-nFormatType: <prop oor:name="Flags">...</prop>
-*/
-static FileFormat aFileFormats[] =
-{
-    { "odp", "impress8", "impress8", "", ODP_FORMAT_TYPE },
-    { "ppt", "MS PowerPoint 97", "impress_MS_PowerPoint_97", "sd", PPT_FORMAT_TYPE },
-    { "pptx", "Impress Office Open XML", "Office Open XML Presentation", "", PPTX_FORMAT_TYPE },
-    { "html", "graphic_HTML", "graphic_HTML", "", HTML_FORMAT_TYPE },
-    { "pdf", "draw_pdf_import", "pdf_Portable_Document_Format", "", PDF_FORMAT_TYPE },
-    { "fodg", "OpenDocument Drawing Flat XML", "draw_ODG_FlatXML", "", FODG_FORMAT_TYPE },
-    { "fodp", "OpenDocument Presentation Flat XML", "impress_ODP_FlatXML", "", FODP_FORMAT_TYPE },
-    { "sxi", "StarOffice XML (Impress)", "impress_StarOffice_XML_Impress", "", SXI_FORMAT_TYPE },
-    { "odg", "draw8", "draw8", "", ODG_FORMAT_TYPE },
-    { "pptm", "Impress MS PowerPoint 2007 XML VBA", "MS PowerPoint 2007 XML VBA", "", PPTM_FORMAT_TYPE },
-    { "potx", "Impress Office Open XML Template", "Office Open XML Presentation Template", "", POTX_FORMAT_TYPE },
-    { nullptr, nullptr, nullptr, nullptr, SfxFilterFlags::NONE }
-};
-
-#define ODP 0
-#define PPT 1
-#define PPTX 2
-#define HTML 3
-#define PDF 4
-#define FODG 5
-#define FODP 6
-#define SXI 7
-#define ODG 8
-#define PPTM 9
-#define POTX 10

```

```

sd::DrawDocShellRef loadURL( const OUString& rURL, sal_Int32 nFormat, std::shared_ptr<SfxAllItemSet> pParams = nullptr )
{
    FileFormat *pFmt = getFormat(nFormat);
    CPPUNIT_ASSERT_MESSAGE( "missing filter info", pFmt->pName != nullptr );
    if ( std::strcmp(pFmt->pName, "odg") == 0 )
    { // Draw
        SotClipboardFormatId nOptions = SotClipboardFormatId::NONE;
        if (pFmt->nFormatType != SfxFilterFlags::NONE)
            nOptions = SotClipboardFormatId::STARDRAW_8;
        auto pFilter = std::make_shared<SfxFilter>(
            OUString::createFromAscii( pFmt->pFilterName ),
            OUString(), pFmt->nFormatType, nOptions,
            OUString::createFromAscii( pFmt->pTypeName ),
            OUString(),
            OUString::createFromAscii( pFmt->pUserData ),
            "private:factory/sdraw*" );
        pFilter->SetVersion(SOFFICE_FILEFORMAT_CURRENT);

        ::sd::DrawDocShellRef xDocShRef = new ::sd::GraphicDocShell(SfxObjectCreateMode::EMBEDDED);
        SfxMedium* pSrcMed = new SfxMedium(rURL, StreamMode::STD_READ, pFilter, std::move(pParams));
        if ( !xDocShRef->DoLoad(pSrcMed) || !xDocShRef.is() )
        {
            if (xDocShRef.is())
                xDocShRef->DoClose();
            CPPUNIT_ASSERT_MESSAGE( OString("failed to load Draw doc" + OUStringToOString( rURL, RTL_TEXTENCODING_U
        )
        CPPUNIT_ASSERT_MESSAGE( "not in destruction", !xDocShRef->IsInDestruction() );
        return xDocShRef;
    }
}
else // Impress
{
    SotClipboardFormatId nOptions = SotClipboardFormatId::NONE;
    if (pFmt->nFormatType != SfxFilterFlags::NONE)
        nOptions = SotClipboardFormatId::STARIMPRESS_8;
    auto pFilter = std::make_shared<SfxFilter>(
        OUString::createFromAscii( pFmt->pFilterName ),
        OUString(), pFmt->nFormatType, nOptions,
        OUString::createFromAscii( pFmt->pTypeName ),
        OUString(),
        OUString::createFromAscii( pFmt->pUserData ),
        "private:factory/simpress*" );
    pFilter->SetVersion(SOFFICE_FILEFORMAT_CURRENT);

    ::sd::DrawDocShellRef xDocShRef = new ::sd::DrawDocShell(SfxObjectCreateMode::EMBEDDED, false, DocumentType
    SfxMedium* pSrcMed = new SfxMedium(rURL, StreamMode::STD_READ, pFilter, std::move(pParams));
    if ( !xDocShRef->DoLoad(pSrcMed) || !xDocShRef.is() )
    {
        if (xDocShRef.is())
            xDocShRef->DoClose();
    }
}

```


Load/Save

- DocShell reference was being passed around all the time.
 - It had to be closed with 'DoClose' when done. Some tests didn't do it.

```
::sd::DrawDocShellRef xDocShRef = loadURL(  
    m_directories.getURLFromSrc(u"/sd/qa/unit/data/odp/transparent_background.odp"), ODP);  
xDocShRef = saveAndReload(xDocShRef.get(), ODP);  
loadFromURL(u"odp/transparent_background.odp");  
saveAndReload("impress8");
```

Load/Save

- Factor out code used for saving everywhere
 - Validate files on export
 - Use 'skipValidation' if not
 - Support for password (it was only available in sw before)

```
- uno::Reference<frame::XStorable> xStorable(getComponent(), uno::UNO_QUERY);  
- uno::Sequence<beans::PropertyValue> aStoreProps = comphelper::InitPropertySequence({  
-     { "FilterName", uno::Any(OUString("writer8")) },  
- });  
- utl::TempFileNamed aTempFile;  
- aTempFile.EnableKillingFile();  
- xStorable->storeToURL(aTempFile.GetURL(), aStoreProps);  
- getComponent()->dispose();  
-  
- getComponent() = loadFromDesktop(aTempFile.GetURL());  
- xTextDocument.set(getComponent(), uno::UNO_QUERY);  
+ saveAndReload("writer8");
```

parseExport

- Sd, sw and sc modules also had each their own methods to parseExport the files instead of using parseExportStream from MacroTest.

```
- ScDocShellRef xDocSh = loadDoc(u"autofilter-colors.", FORMAT_XLSX);  
- std::shared_ptr<utl::TempFileNamed> pXPathFile  
-     = ScBootstrapFixture::exportTo(*xDocSh, FORMAT_XLSX);  
- xmlDocUniquePtr pTable1  
-     = XPathHelper::parseExport(pXPathFile, m_xSFactory, "xl/tables/table1.xml");  
+ createScDoc("xlsx/autofilter-colors.xlsx");  
+ save("Calc Office Open XML");  
+ xmlDocUniquePtr pTable1 = parseExport("xl/tables/table1.xml");
```

Other clean ups

- Drop duplicate operators

```
-std::ostream& operator<<(std::ostream& rStrm, const Color& rColor)
-{
    rStrm << "Color: R:" << static_cast<int>(rColor.GetRed()) << " G:" << static_cast<int>(rColor.GetGreen()) << " B: " << static_cast<int>(rColor.GetBlue());
    return rStrm;
-}
```

- Fix CppunitTest_sc_opencl_test
 - Compares calculation with and without opencl enabled
- Drop custom asserts

[illegible]

Thanks a lot!

Xisco Fauli

xiscofauli@libreoffice.org