



Things you can test in a UITest

Xisco Fauli

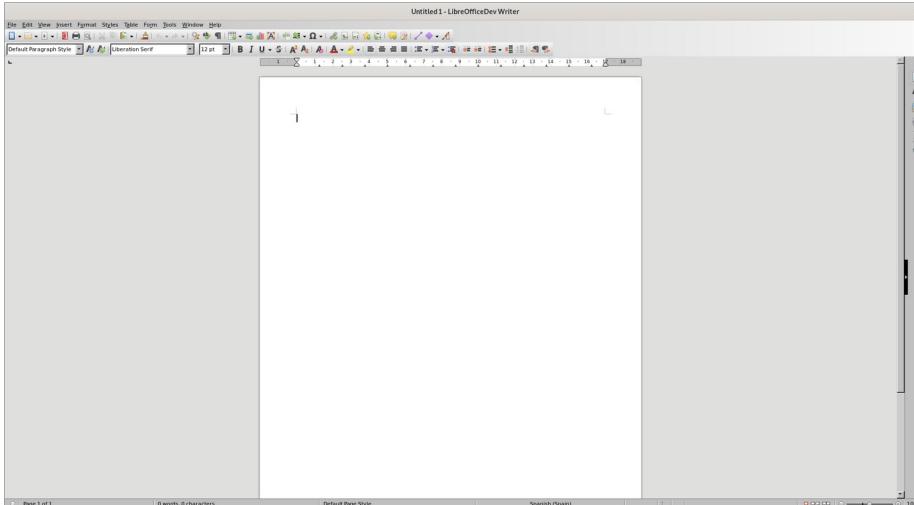


Create empty document

```
with self.ui_test.create_doc_in_start_center("writer") as document:
```

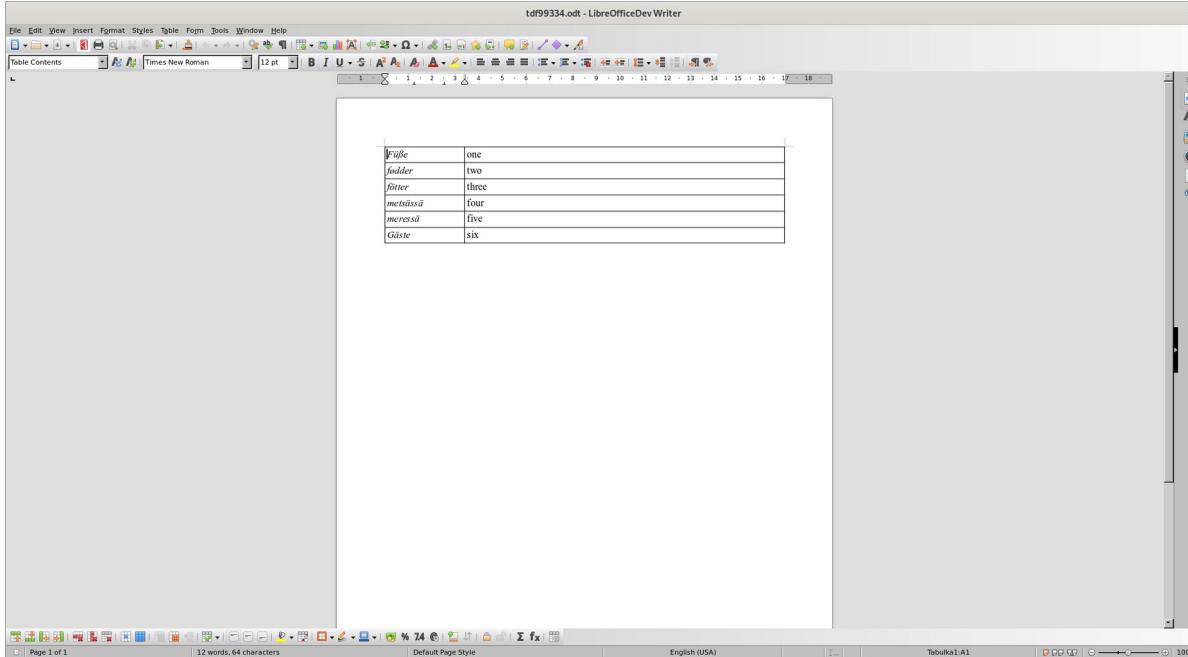
```
with self.ui_test.create_doc_in_start_center("calc") as document:
```

```
with self.ui_test.create_doc_in_start_center("impress") as document:
```



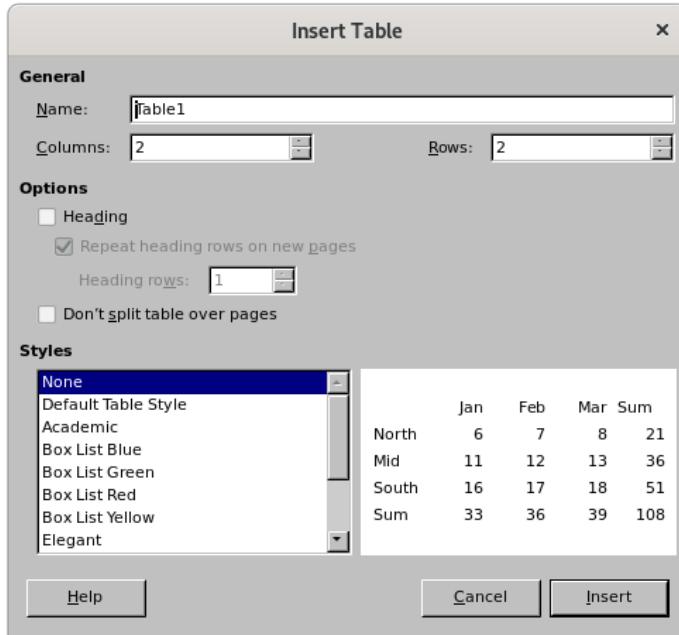
load document

```
from uitest.uihelper.common import get_url_for_data_file  
  
with self.ui_test.load_file(get_url_for_data_file("tdf99334.odt")) as document:
```



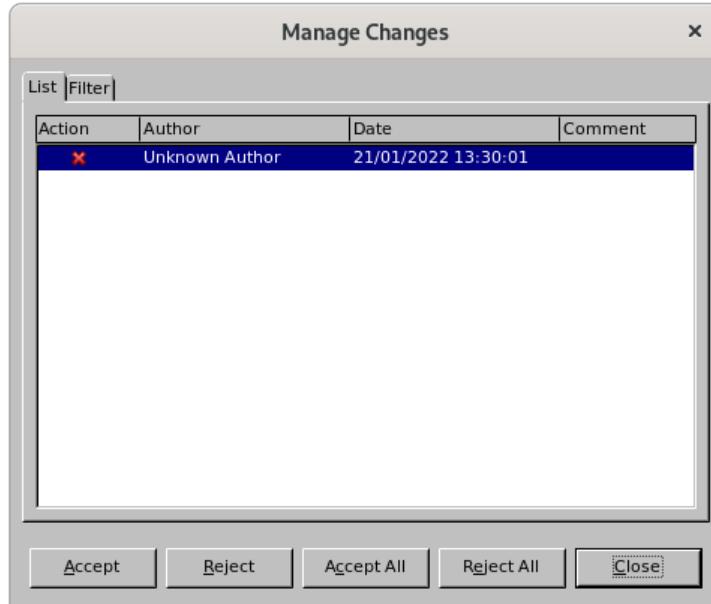
Execute dialog

```
with self.ui_test.execute_dialog_through_command(".uno:InsertTable") as xDialog:
```



Execute modeless dialog

```
with self.ui_test.execute_modeless_dialog_through_command(".uno:AcceptTrackedChanges",  
close_button="close"):
```



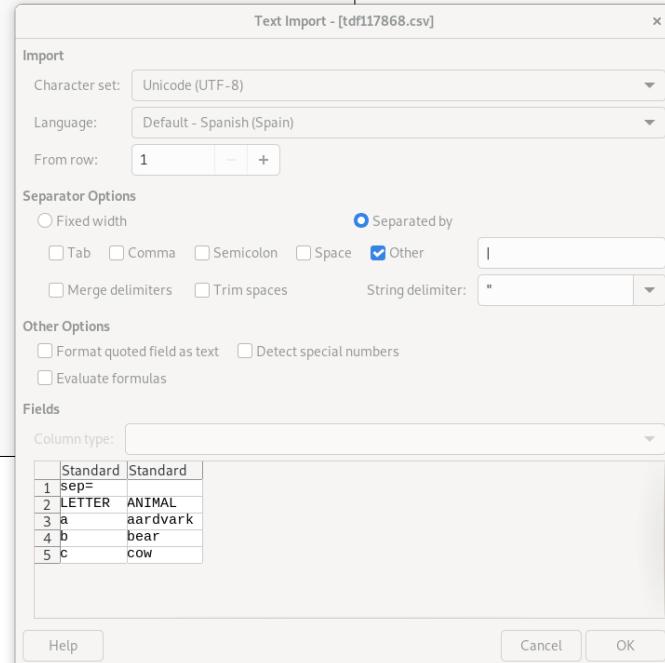
load document with dialog

```
from uitest.uihelper.common import get_url_for_data_file

with self.ui_test.execute_dialog_through_command(".uno:Open", close_button="") as
xOpenDialog:
    xFileName = xOpenDialog.getChild("file_name")
    xFileName.executeAction("TYPE", mkPropertyValues({"TEXT":get_url_for_data_file("tdf145326.odt")}))
```



```
xOpenBtn = xOpenDialog.getChild("open")
with self.ui_test.wait_until_component_loaded():
    with self.ui_test.execute_blocking_action(xOpenBtn.executeAction,
        args=('CLICK',()), close_button="yes"):
        pass
```



Save document and reload

```
from org.libreoffice.unotest import systemPathToFileUrl
from tempfile import TemporaryDirectory
import os.path

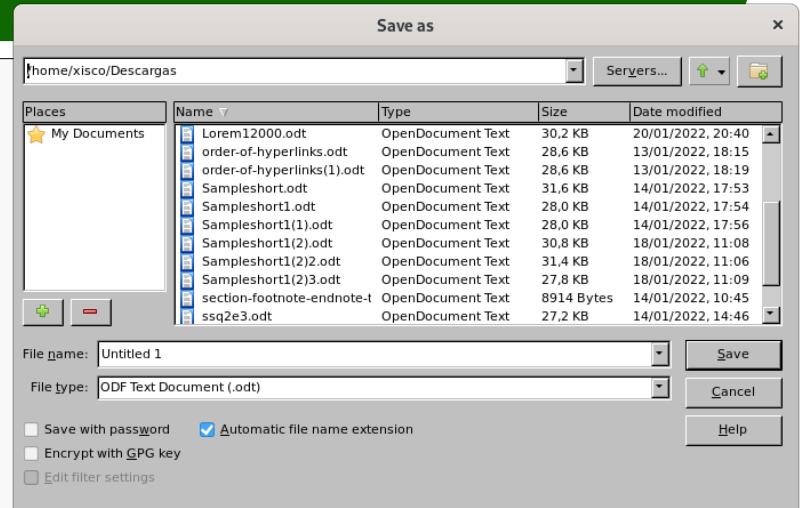
with TemporaryDirectory() as tempdir:
    xFilePath = os.path.join(tempdir, "tdf119661-tmp.odt")

    with self.ui_test.create_doc_in_start_center("writer"):

        with self.ui_test.execute_dialog_through_command(".uno:SaveAs",
close_button="open") as xDialog:

            xFileName = xDialog.getChild("file_name")
            xFileName.executeAction("TYPE", mkPropertyValues({"KEYCODE":"CTRL+A"}))
            xFileName.executeAction("TYPE", mkPropertyValues({"KEYCODE":"BACKSPACE"}))
            xFileName.executeAction("TYPE", mkPropertyValues({"TEXT": xFilePath}))

    with self.ui_test.load_file(systemPathToFileUrl(xFilePath)):
```



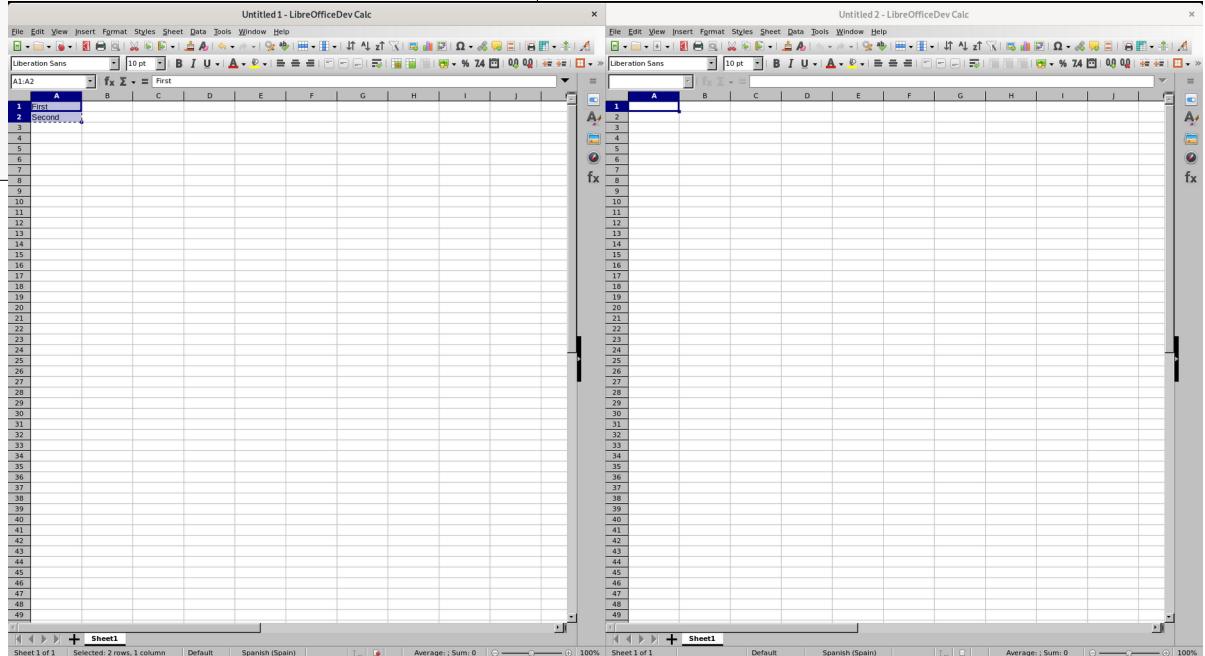
Different documents

```
with self.ui_test.create_doc_in_start_center("calc") as document:
```

```
    with self.ui_test.load_empty_file("calc") as document2:
```

```
        frames = self.ui_test.get_frames()  
        # switch view to the first document  
        frames[0].activate()
```

```
        # switch view back to the second document  
        frames[1].activate()
```



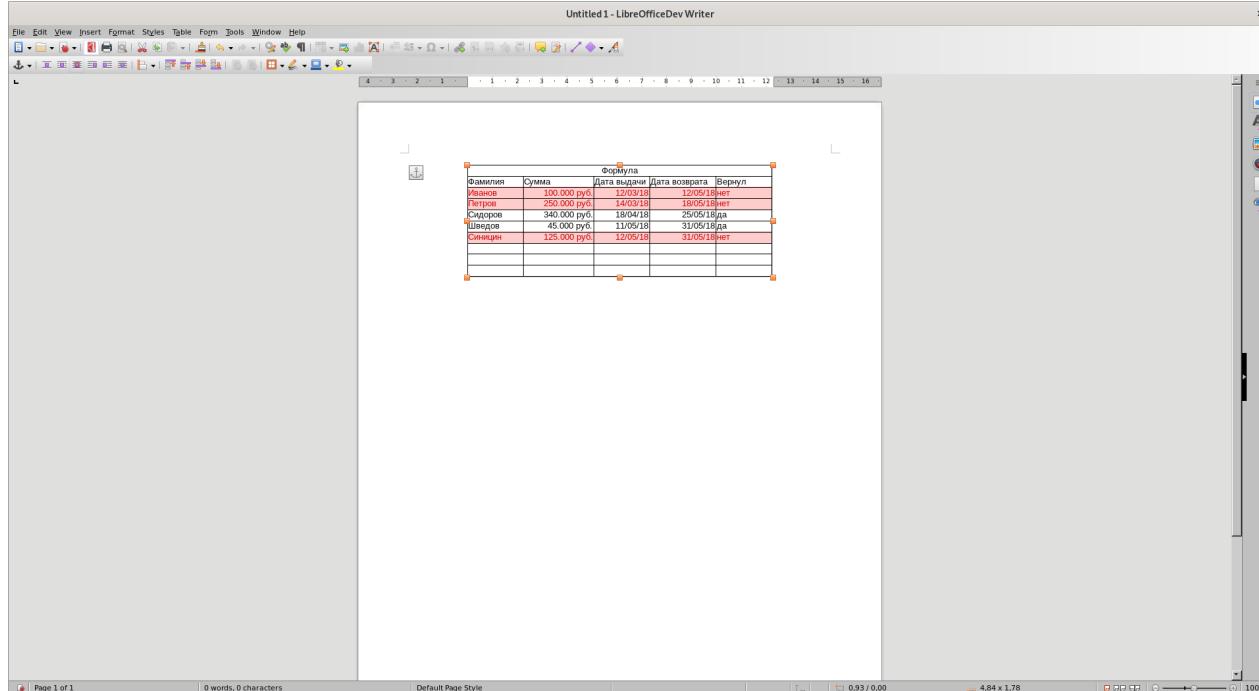
Interaction between components

```
with self.ui_test.create_doc_in_start_center("calc") as document:
```

```
...
```

```
with self.ui_test.load_empty_file("writer") as writer_document:
```

```
...
```



Sidebar

```
xWriterDoc = self.xUITest.getTopFocusWindow()
xWriterEdit = xWriterDoc.getChild("writer_edit")

self.xUITest.executeCommand(".uno:Sidebar")
xWriterEdit.executeAction("SIDEBAR", mkPropertyValues(
    {"PANEL": "InspectorTextPanel"}))

# Close sidebar after using it
self.xUITest.executeCommand(".uno:Sidebar")
```



Style Inspector

Properties	Values
Paragraph Styles	
Default Paragraph Style	
Border Distance	0
Bottom Border	
Color	0x0
Line Width	0
Line Style	0
Line Distance	0
Inner Line Width	0
Outer Line Width	0
Bottom Border Distance	0
Category	0
Char Auto Kerning	True
Char Back Color	0x-1
Char Back Transparent	True
Char Border Distance	0
Char Bottom Border	
Color	0x0
Line Width	0
Line Style	0
Line Distance	0
Inner Line Width	0
Outer Line Width	0
Char Bottom Border Distance	0
Char Case Map	0
Char Color	0x-1
Char Combine is On	False
Char Contoured	False
Char Crossed Out	False
Char Difference Height	0
Char Difference Height 0	0
Char Difference Height 1	0
Char Emphasis	0
Char Escapement	0
Char Escapement Height	100
Char Flash	False
Char Font Char Set	76
Char Font Char Set Asian	76
Char Font Char Set Complex	76
Char Font Family	3
Char Font Family Asian	6
Char Font Family Complex	6
Char Font Name	Liber
Char Font Name Asian	NSim
Char Font Name Complex	Alba
Char Font Pitch	2
Char Font Pitch Asian	2
Char Font Pitch Complex	2
Char Height	12
Char Height Asian	10.5
Char Height Complex	12
Char Hidden	False
Char Highlight	-1

Navigator

```
xWriterDoc = self.xUITest.getTopFocusWindow()
xWriterEdit = xWriterDoc.getChild("writer_edit")

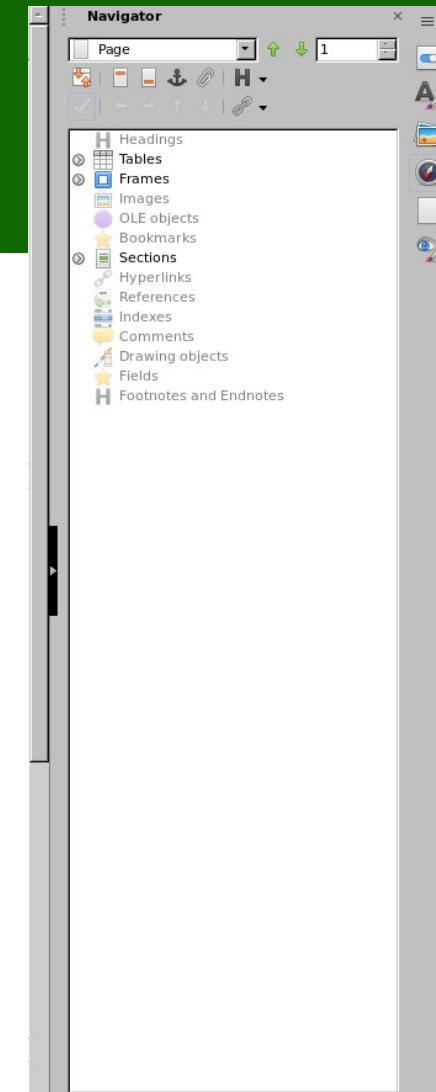
self.xUITest.executeCommand(".uno:Sidebar")

xWriterEdit.executeAction("SIDEBAR",
mkPropertyValues({"PANEL": "SwNavigatorPanel"}))

# wait until the navigator panel is available
xNavigatorPanel =
self.ui_test.wait_until_child_is_available('NavigatorPanel')

xContentTree = xNavigatorPanel.getChild("contenttree")

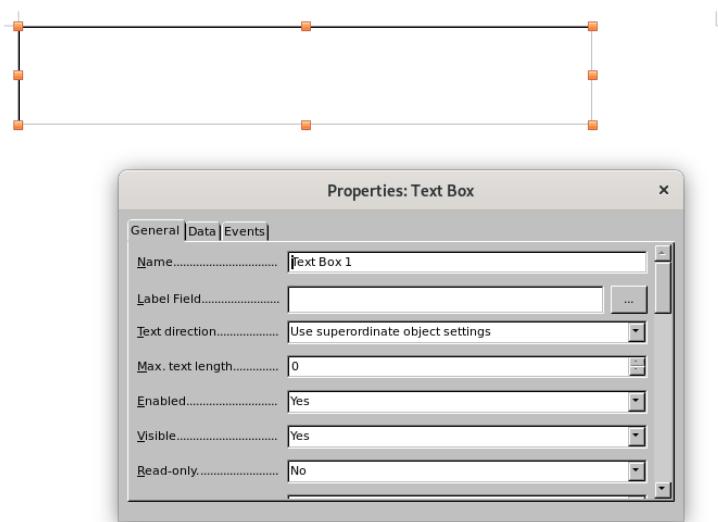
# Close sidebar after using it
self.xUITest.executeCommand(".uno:Sidebar")
```



Forms

```
self.xUITest.executeCommand(".uno:JumpToNextFrame")
```

```
with self.ui_test.execute_modeless_dialog_through_command(".uno:ControlProperties",  
close_button=""):  
    xChild = self.ui_test.wait_until_child_is_available('listbox-Text type'))
```

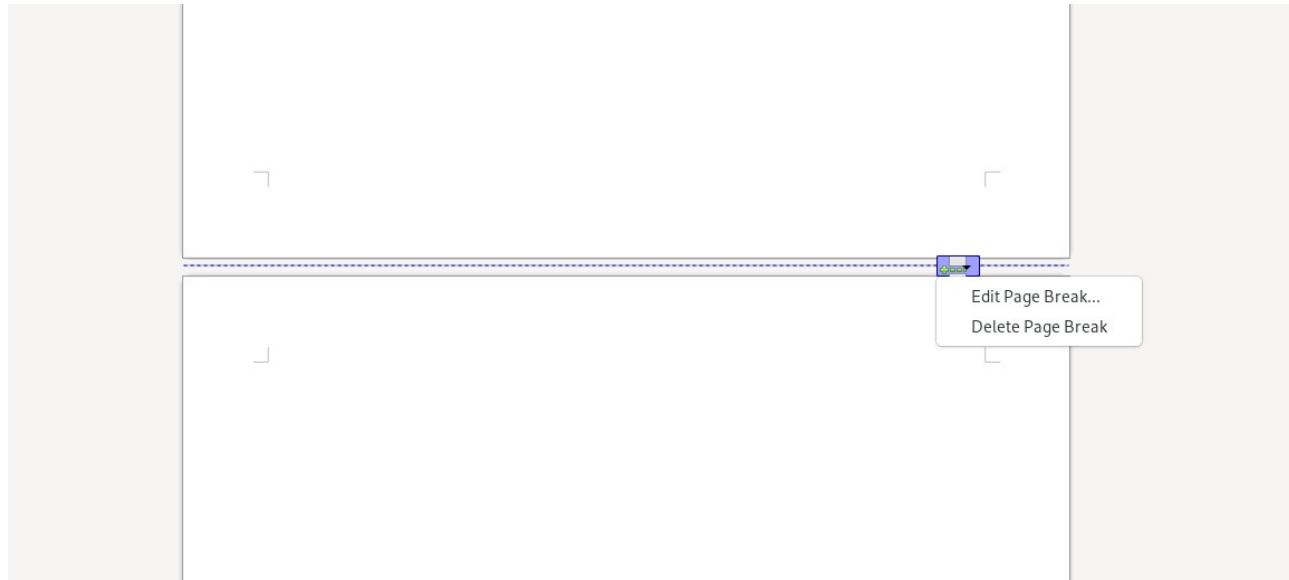


Pagebreak

```
self.xUITest.executeCommand(".uno:InsertPagebreak")
```

```
xPageBreak = self.ui_test.wait_until_child_is_available('PageBreak')
```

```
with self.ui_test.execute_dialog_through_action(xPageBreak, "EDIT") as xDialog:
```

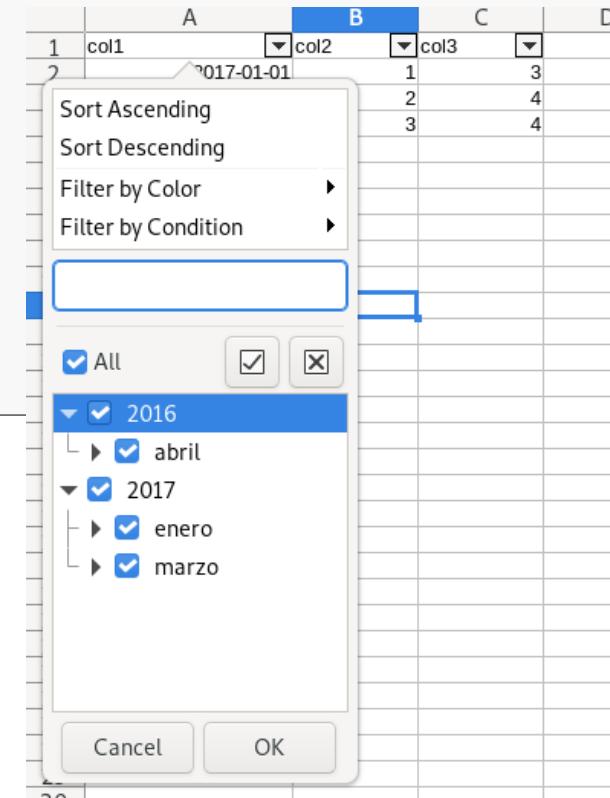


Autofilter

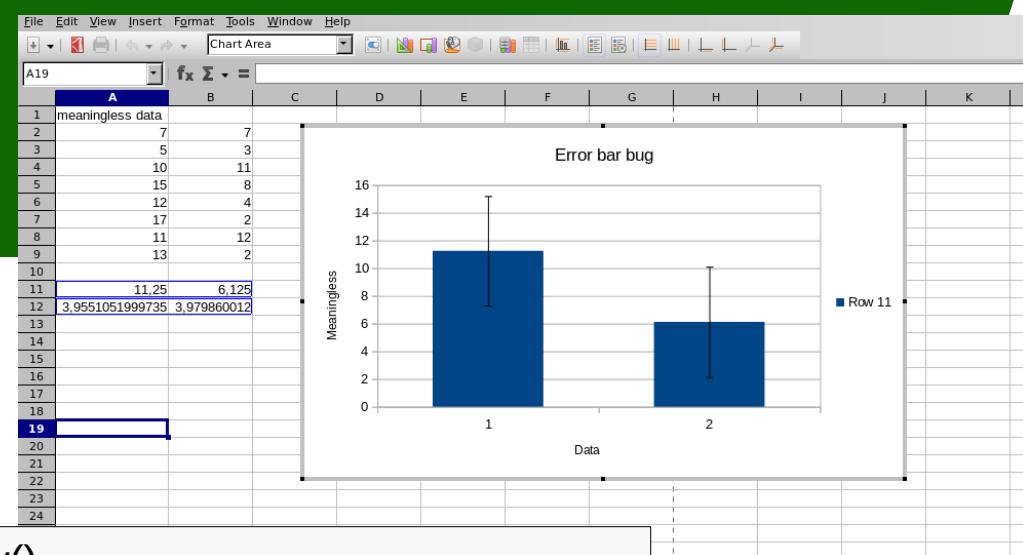
```
xGridWin = self.xUITest.getTopFocusWindow().getChild("grid_window")
xGridWin.executeAction("LAUNCH", mkPropertyValues({"AUTOFILTER": "", "COL": "0", "ROW": "0"}))

xFloatWindow = self.xUITest.getFloatWindow()
xCheckListMenu = xFloatWindow.getChild("FilterDropDown")
xTreeList = xCheckListMenu.getChild("check_tree_box")
xFirstEntry = xTreeList.getChild("0")
xFirstEntry.executeAction("CLICK", tuple())

xOkBtn = xFloatWindow.getChild("ok")
xOkBtn.executeAction("CLICK", tuple())
```



Charts



```
xCalcDoc = self.xUITest.getTopFocusWindow()  
gridwin = xCalcDoc.getChild("grid_window")
```

```
gridwin.executeAction("SELECT", mkPropertyValues({"OBJECT": "Object 1"}))  
gridwin.executeAction("ACTIVATE", tuple())
```

```
xChartMainTop = self.xUITest.getTopFocusWindow()  
xChartMain = xChartMainTop.getChild("chart_window")
```

```
xSeriesObj = xChartMain.getChild("CID/D=0:CS=0:CT=0:Series=0")  
with self.ui_test.execute_dialog_through_action(xSeriesObj, "COMMAND",  
mkPropertyValues({"COMMAND": "FormatYErrorBars"})) as xDialog:
```



Thanks

Xisco Fauli
The Document Foundation
xiscofauli@libreoffice.org

