

# LibreOffice QA

## How to write your first test

**Xisco Fauli**

[xiscofauli@libreoffice.org](mailto:xiscofauli@libreoffice.org)

# Motivation

- A bug fix without a unit test WILL GET BROKEN AGAIN in the next release.
- A bug fix with a unit test WILL REMAIN FIXED FOREVER.

**Which one will be your choice?**

From Kohei's talk (slide 26):

<https://conference.libreoffice.org/assets/Conference/Bern/slides/KoheiYoshida.pdf>

# Agenda

- Python – UI Testing
- CppUnit Testing
  - Import/Export
  - assertXPath
  - Layout

# Prerequisites

- LibreOffice source code
- Some Python/C++ skills
- Some git skills
- Gerrit account
- Desire to learn ;-)

# Disclaimer

This presentation will mainly talk about Writer tests.

# Python Uitestng - Info

- Inherit from unittest.TestCase

<https://docs.python.org/3/library/unittest.html>

- Used for testing the UI
- ~600 existing tests
- In writer, they are under sw/qa/uitest/
- Slower than CppUnit tests
- Linux only
- <https://wiki.documentfoundation.org/Development/UITests>

# Python UTesting – First test

1. *LO\_COLLECT\_UIINFO="test.log" SAL\_USE\_VCLPLUGIN=gen  
instdir/program/soffice*
2. Write the test. Some useful prints:
  - *print(xFoo.getChildren())*
  - *print(get\_state\_as\_dict(xFoo))*
  - *print(dir(xFoo))*
3. Run the script and visualize it  
[https://wiki.documentfoundation.org/Development/UITests#Running\\_the\\_test](https://wiki.documentfoundation.org/Development/UITests#Running_the_test)
4. Repeat
5. Commit

# CppUnit Testing

- In Writer, they are under `sw/qa/extras/` divided in subfolders.
  - ODT import → `sw/qa/extras/odfimport/`
  - OOXML export → `sw/qa/extras/ooxmlexport/`
  - Layout → `sw/qa/extras/layout/`
- ~3000 existing tests in `sw/qa/extras/`
- `sw/qa/extras/README`
- [https://wiki.documentfoundation.org/Development/Cpp\\_Unit\\_Tests](https://wiki.documentfoundation.org/Development/Cpp_Unit_Tests)



# CppUnit Testing - Run

- `make <module>`

*make CppunitTest\_sw\_ooxmlexport16*

- `CPPUNIT_TEST_NAME="testName" make -sr <module>`

*CPPUNIT\_TEST\_NAME="testTdf138953" make -sr CppunitTest\_sw\_ooxmlexport16*

# CppUnit Testing – Import/Export

- **Import Tests:** load the specified file to 'mxComponent', which represents the UNO model of the document
- **Export Tests:** Same as import tests but the document is exported and imported again afterwards
- **Export Only tests:** Only exports

# CppUnit Testing – assertXPath

- **Import Tests**

```
xmlDocUniquePtr pXmlDoc = parseLayoutDump();
```

- **Export Tests**

```
xmlDocUniquePtr pXmlDoc = parseExport();
```

Document Analyzers:

[https://wiki.documentfoundation.org/Development/Analyzing\\_Documents](https://wiki.documentfoundation.org/Development/Analyzing_Documents)

# CppUnit Testing – Layout

- Dump layout to XML file with **--convert-to 'xml:writer\_layout\_dump'**

<http://bosdonnat.fr/testing-writer-documents-rendering.html>

- Then use `assertXPath`

# Write your first test today!

- Python Uitestng

[https://wiki.documentfoundation.org/Development/UI\Tests#Finding\\_bugs](https://wiki.documentfoundation.org/Development/UI\Tests#Finding_bugs)

- CppUnit Tests

<https://wiki.documentfoundation.org/MissingUnitTests>

**Thanks a lot!**

**Xisco Fauli**

[xiscofauli@libreoffice.org](mailto:xiscofauli@libreoffice.org)