Python embedded in LibreOffice or OpenOffice documents

Hamilton Python Users Group
Ian Stewart
11-Oct-2021

Repositories:
https://github.com/irsbugs/LO-OO-Macro-Programming
https://github.com/HamPUG/meetings/tree/master/2021/2021-10-11

Python embedded in LO/OO documents

- LibreOffice and OpenOffice applications default to allowing BASIC macros to be stored within a Document.

- Python macros may also be stored within a document.

- Inserting Python macros is made easier using the LibreOffice addon, “Alternative Python Script Organizer”, ~~ASPO~~ APSO.

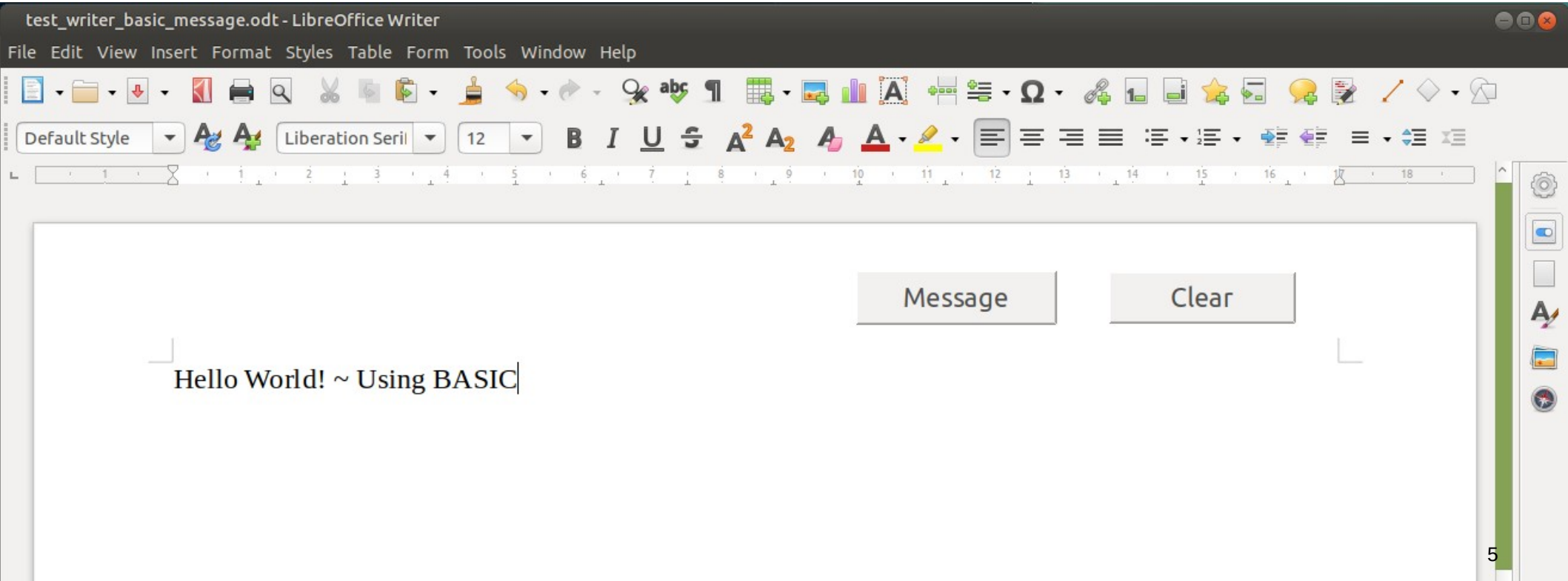
Previous Presentation on BASIC in LO/OO

- WLUG meeting on 23 Aug 2021 presented BASIC in LO/OO.
- WLUG repository posting presentation slides and documents...
  https://github.com/WLUG/meetings/tree/master/2021/2021-08-23
- Documents included are:
  - Simple Writer document.
  - Draw document simulating piling for a house
  - Calc document providing modelling the amortization of a loan.
- The presentation was at a virtual meeting that was video recorded. Playback of the video is available at:
  https://bbb.nzoss.nz/playback/presentation/2.0/playback.html?meetingId=b9e64e4bd74566f13a48d6a5940f3c929982b914-1629703025339

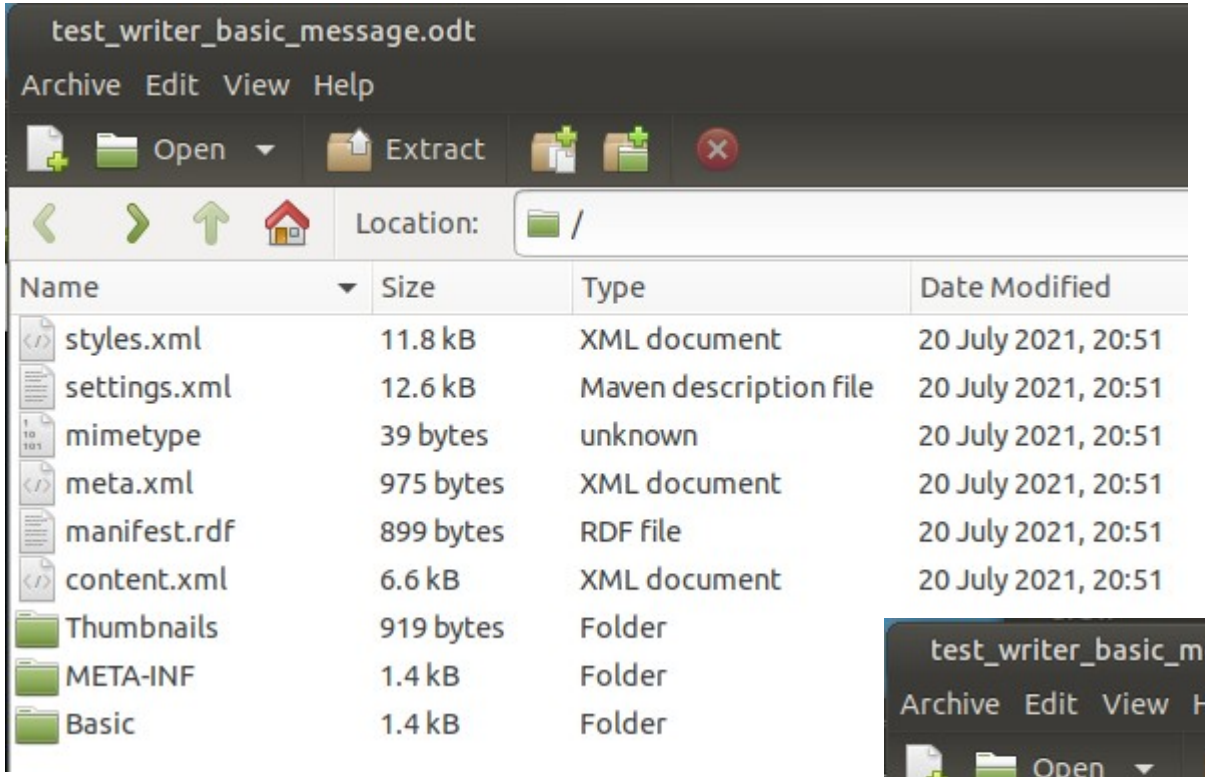
BASIC embedded in LO/OO documents (default)

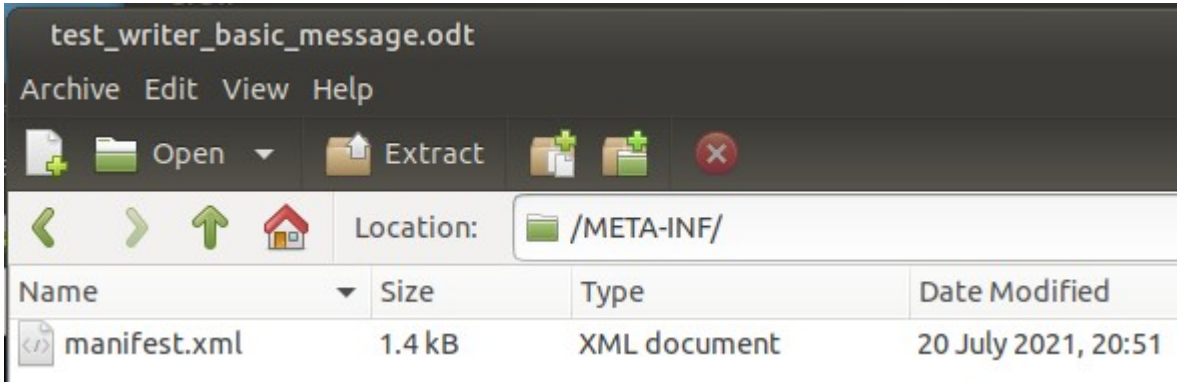
BASIC embedded in LO/OO documents

Writer document with two push buttons that execute BASIC code...



Unzipping of Writer document “test_writer_basic_message.odt”...





“manifest.xml” lists BASIC scripts are included...

manifest.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <manifest:manifest xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0" manifest:version="1.2"
   xmlns:loext="urn:org:documentfoundation:names:experimental:office:xmlns:loext:1.0">
3   <manifest:file-entry manifest:full-path="/" manifest:version="1.2" manifest:media-type="application/-
   vnd.oasis.opendocument.text"/>
4   <manifest:file-entry manifest:full-path="Basic/Standard/Module1.xml" manifest:media-type="text/xml"/>
5   <manifest:file-entry manifest:full-path="Basic/Standard/script-lb.xml" manifest:media-type="text/xml"/>
6   <manifest:file-entry manifest:full-path="Basic/script-lc.xml" manifest:media-type="text/xml"/>
7   <manifest:file-entry manifest:full-path="Configurations2/" manifest:media-type="application/vnd.sun.xml.ui.configuration"/>
8   <manifest:file-entry manifest:full-path="manifest.rdf" manifest:media-type="application/rdf+xml"/>
9   <manifest:file-entry manifest:full-path="meta.xml" manifest:media-type="text/xml"/>
10  <manifest:file-entry manifest:full-path="settings.xml" manifest:media-type="text/xml"/>
11  <manifest:file-entry manifest:full-path="Thumbnails/thumbnail.png" manifest:media-type="image/png"/>
12  <manifest:file-entry manifest:full-path="styles.xml" manifest:media-type="text/xml"/>
13  <manifest:file-entry manifest:full-path="content.xml" manifest:media-type="text/xml"/>
14 </manifest:manifest>
```

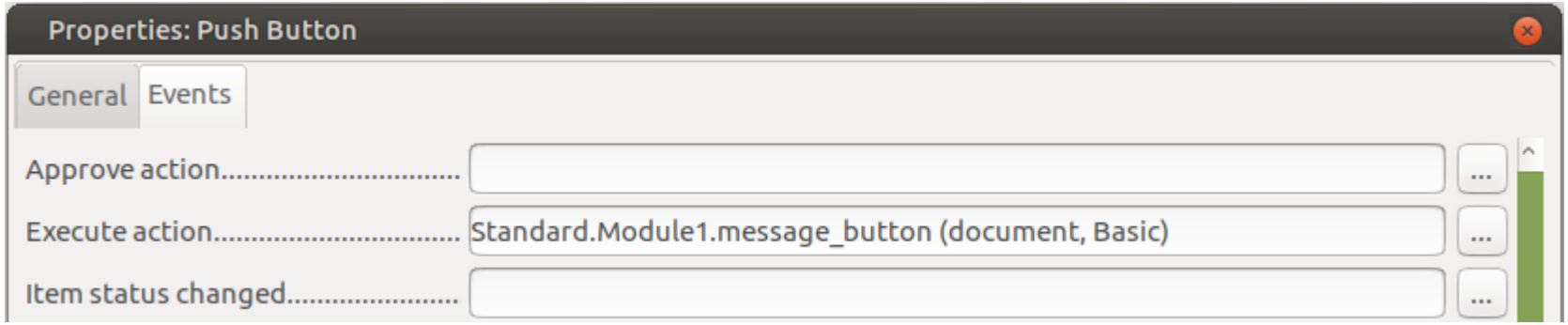
```
"Basic/Standard/Module1.xml" manifest:media-type="text/xml"/>
"Basic/Standard/script-lb.xml" manifest:media-type="text/xml"/>
"Basic/script-lc.xml" manifest:media-type="text/xml"/>
```

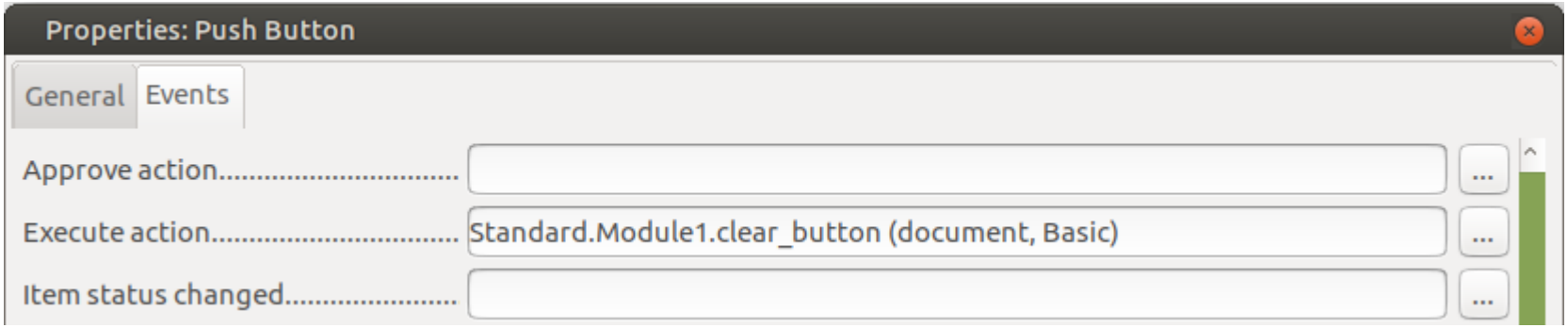
BASIC/Standard/Module1 has the BASIC script for each push button to execute...

Module1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE script:module PUBLIC "-//OpenOffice.org//DTD OfficeDocument 1.0//EN" "module.dtd">
<script:module xmlns:script="http://openoffice.org/2000/script" script:name="Module1"
script:language="StarBasic" script:moduleType="normal">REM  *****  BASIC  *****

Sub Main

End Sub

sub message_button(button)
    doc = ThisComponent
    doc.getText().setString(&quot;Hello World! ~ Using BASIC&quot;)
end sub

sub clear_button(button)
    doc = ThisComponent
    doc.getText().setString(&quot;&quot;)
end sub

</script:module>
```

“Execute Action” has description of path to push button BASIC subroutines for “Message” and “Clear”...





BASIC IDE: Tools --> Macros --> Edit Macros.

test_writer_basic_message.odt.Standard - LibreOffice Basic

File Edit View Run Dialog Tools Window Help

[test_writer_basic_message.oc

Object Catalog

- My Macros & Dialogs
  - MRILib
  - Standard
- LibreOffice Macros & Dialogs
- Python Embedded Presentation.odp
- test_writer_basic_message.odt
  - Standard
    - Module1
      - Main
      - message_button
      - clear_button

```
REM  *****  BASIC  *****

sub message_button(button)
    doc = ThisComponent
    doc.getText().setString("Hello World! ~ Using BASIC")
end sub

sub clear_button(button)
    doc = ThisComponent
    doc.getText().setString("")
end sub
```

Watch:

Variable	Value	Type

Calls:

Module1

Find Find All Match Case

test_writer_basic_message.odt.Standard.Module1.clear_button Ln 11, Col 8

Python embedded in LO/OO documents

But first: Python NOT embedded in LO/OO documents

LibreOffice and OpenOffice applications with Python macros can be stored in:

- My Macros (In User account space)

```
ian@leno:~/.config/libreoffice/4/user/Scripts/python$ ls -l
-rw-rw-r-- 1 ian ian 25129 Oct  8 20:58 draw_uno_plan.py
```

- LibreOffice Macros (In System-wide space)

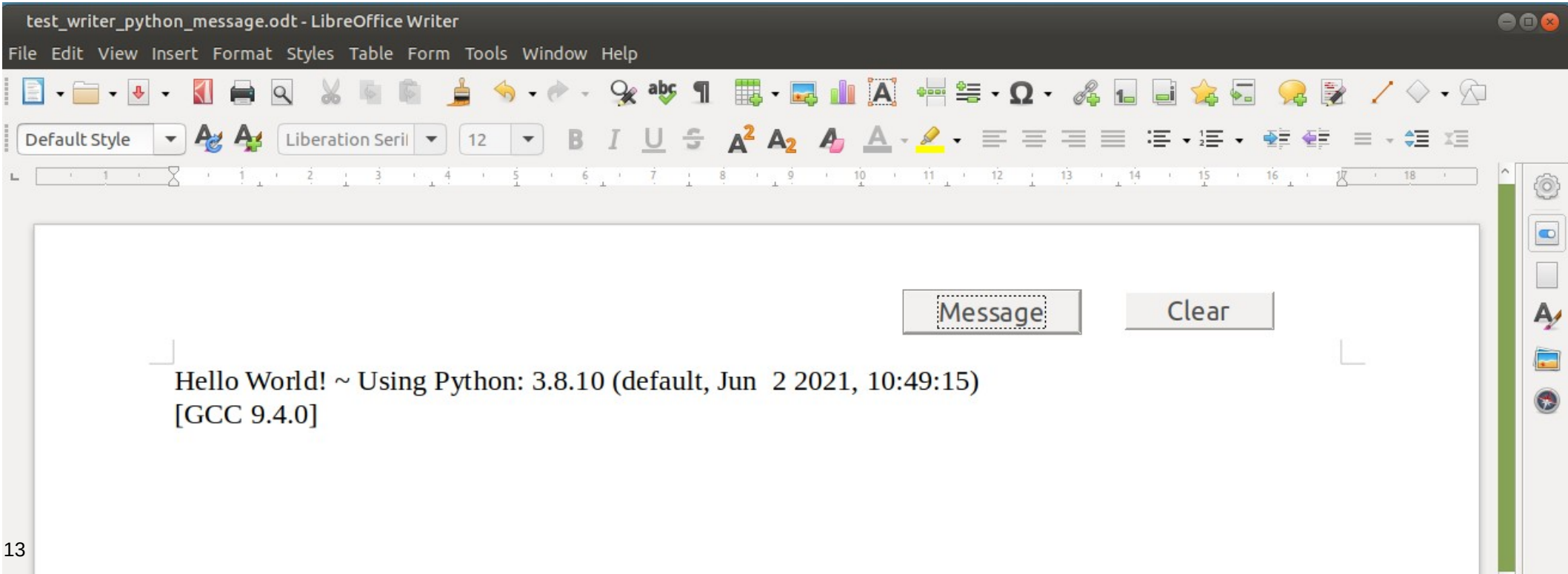
```
ian@leno:/usr/lib/libreoffice/share/Scripts/python$ ls -l
-rw-r--r-- 1 root root 3384 Oct  8  2020 Capitalise.py
-rw-r--r-- 1 root root 1589 Oct  8  2020 HelloWorld.py
-rw-r--r-- 1 root root 2406 Oct  8  2020 InsertText.py
-rw-r--r-- 1 root root 2090 Oct  8  2020 NamedRanges.py
drwxr-xr-x 2 root root 4096 Apr 23 17:26 pythonSamples
-rw-r--r-- 1 root root  634 Oct  8  2020 SetCellColor.py
```

- Need a PyUno bridge from Python program into LibreOffice:

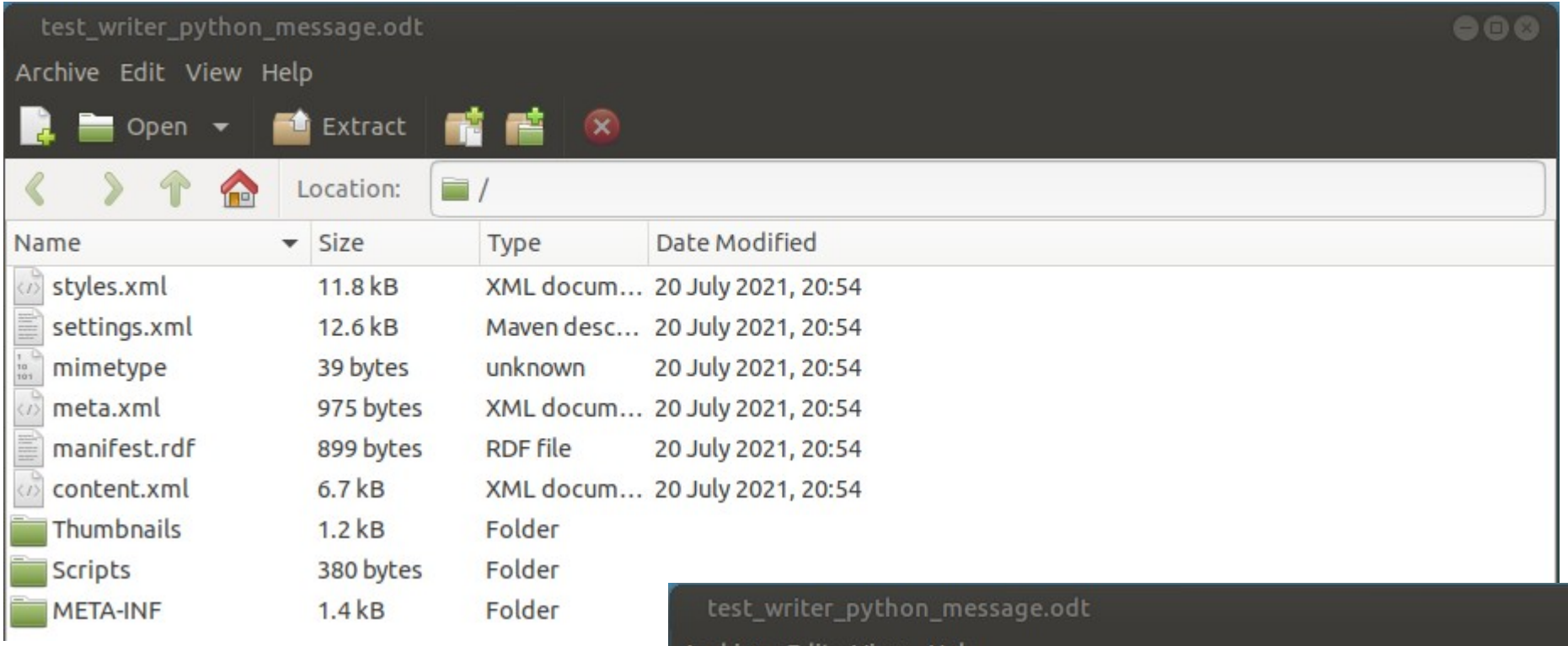
```
$ libreoffice --draw --accept="socket,host=localhost,port=2002;urp;StarOffice.ServiceManager"
```

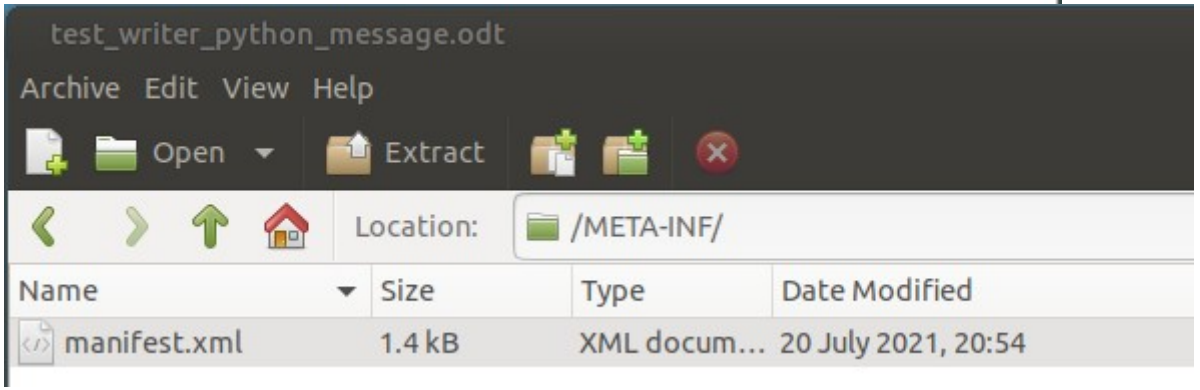
Python embedded in LO/OO documents

LibreOffice and OpenOffice applications with Python macros stored in a Document.



Unzipping of Writer document to view the embedded Python macros





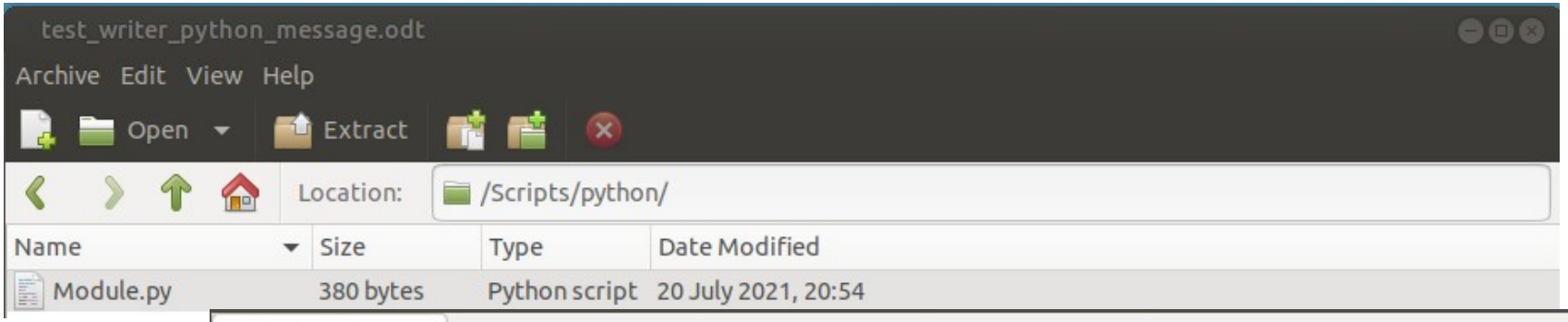
“manifest.xml” file contains references to the Python files.

manifest.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0" manifest:version="1.2" xmlns:loext="urn:org:documentfoundation:names:experimental:office:xmlns:loext:1.0">
 <manifest:file-entry manifest:full-path="/" manifest:version="1.2" manifest:media-type="application/-vnd.oasis.opendocument.text"/>
 <manifest:file-entry manifest:full-path="Configurations2/" manifest:media-type="application/-vnd.sun.xml.ui.configuration"/>
 <manifest:file-entry manifest:full-path="Scripts/python/Module.py" manifest:media-type=""/>
 <manifest:file-entry manifest:full-path="Scripts/python/" manifest:media-type="application/binary"/>
 <manifest:file-entry manifest:full-path="Scripts/" manifest:media-type="application/binary"/>
 <manifest:file-entry manifest:full-path="manifest.rdf" manifest:media-type="application/rdf+xml"/>
 <manifest:file-entry manifest:full-path="meta.xml" manifest:media-type="text/xml"/>
 <manifest:file-entry manifest:full-path="settings.xml" manifest:media-type="text/xml"/>
 <manifest:file-entry manifest:full-path="Thumbnails/thumbnail.png" manifest:media-type="image/png"/>
 <manifest:file-entry manifest:full-path="styles.xml" manifest:media-type="text/xml"/>
 <manifest:file-entry manifest:full-path="content.xml" manifest:media-type="text/xml"/>
</manifest:manifest>
```

```xml
"Scripts/python/Module.py" manifest:media-type=""/>
"Scripts/python/" manifest:media-type="application/binary"/>
"Scripts/" manifest:media-type="application/binary"/>
```

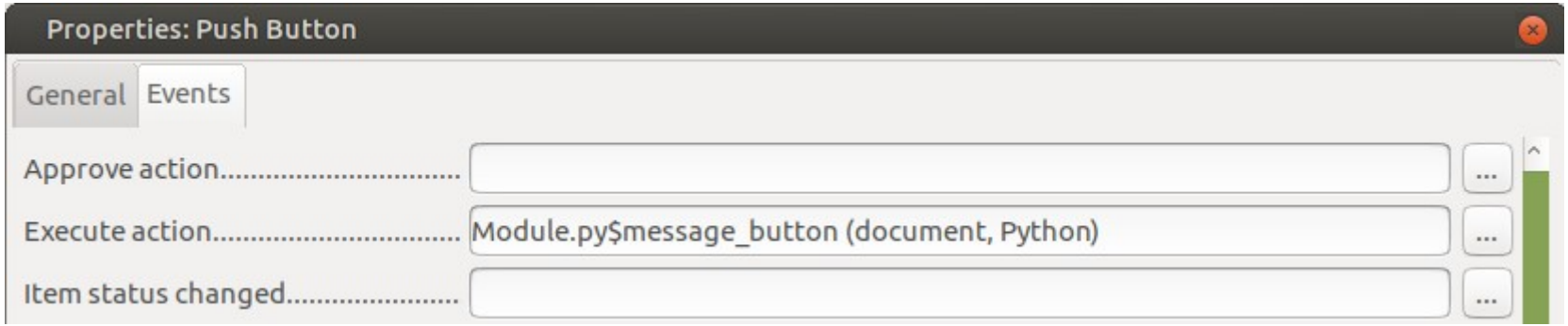
Python script in “Module.py”, in “test_writer_python_message.odt” file...

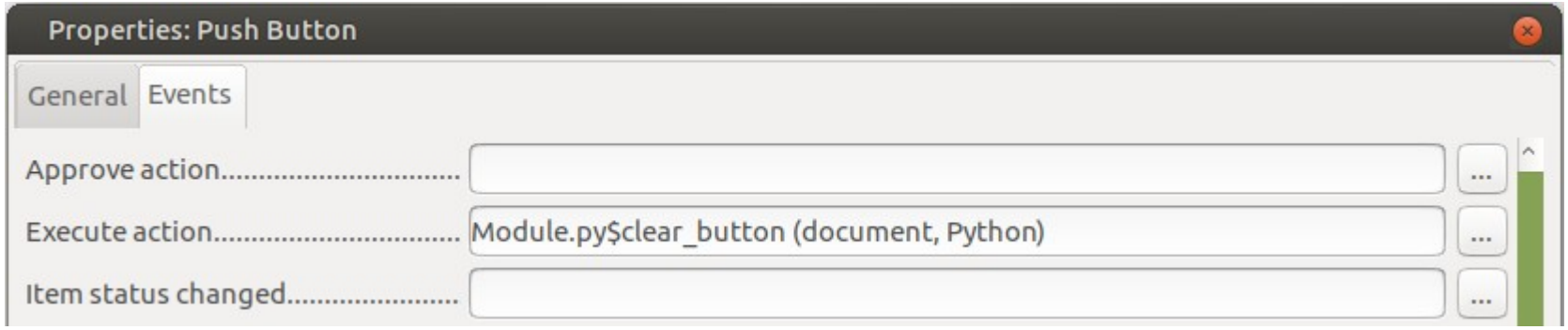


Module.py

```python
# coding: utf-8
from __future__ import unicode_literals

import uno
import sys

def message_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("Hello World! ~ Using Python: " + sys.version)

def clear_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("")

g_exportedScripts = message_button, clear_button,
```

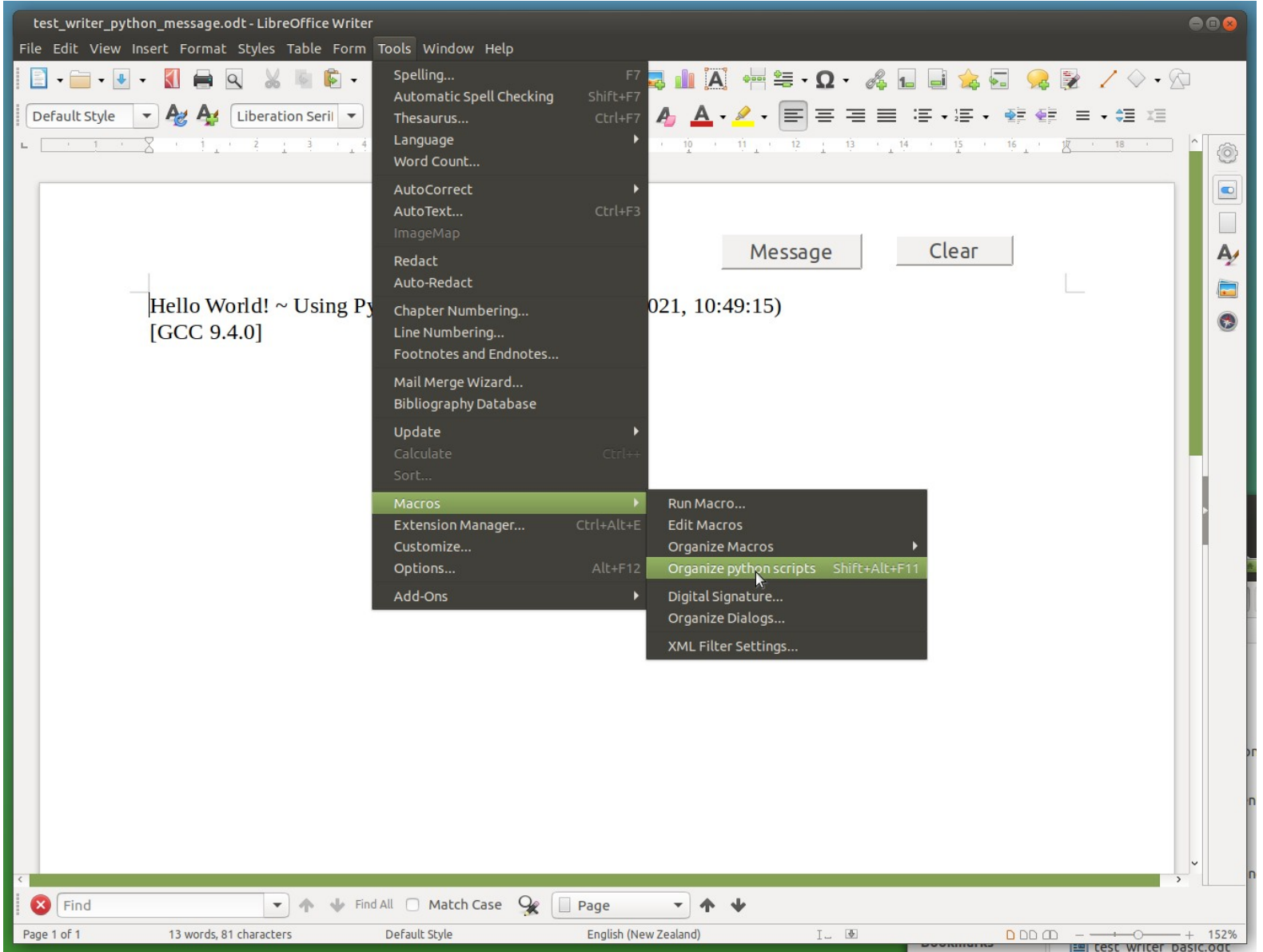
Push button events for “message” and “clear” buttons...



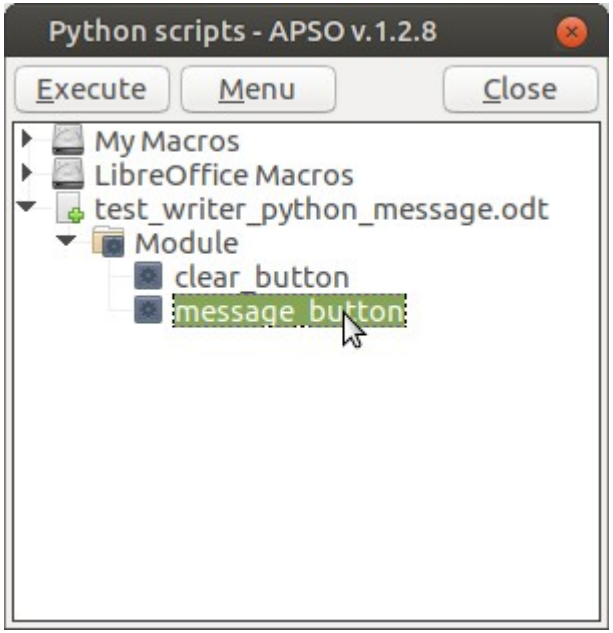


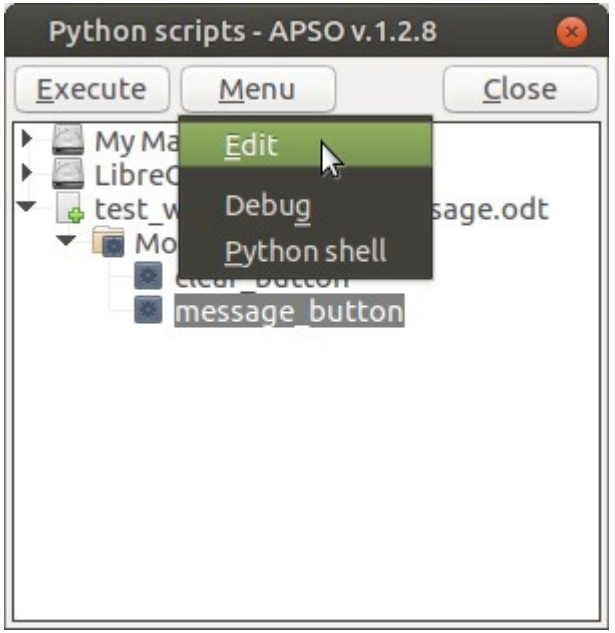
Alternative Python Scripts Organizer (APSO)

APSO ~ Alternative Python Scripts Organizer



APSO


Python scripts - APSO v.1.2.8
Execute
Menu
Close
My Macros
LibreOffice Macros
test_writer_python_message.odt
Module
clear_button
message_button


Python scripts - APSO v.1.2.8
Execute
Menu
Close
Edit
Debug
Python shell
message_button

Example of “Geany” as the Python IDE.

Module.py - /tmp/lu2365ckm6h7.tmp/10/Scripts/python - Geany

File Edit Search View Document Project Build Tools Help

Symbols | Documents

- ~
- ~/dev/alpha
- ~/dev/alpha/alpha_instruction_builder
- ~/dev/alpha/alpha_in...ction_builder/opcodes
- ~/dev/alpha/treeview
- ~/dev/builder
- /tmp/lu2365ckm6h7.tmp/10/Scripts/python
  - Module.py

get_instructions_no_r31.py | alpha_gui_geany_4.py | aib_v1.py | Module.py

```python
# coding: utf-8
from __future__ import unicode_literals

import uno
import sys

def message_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("Hello World! ~ Using Python: " + sys.version)

def clear_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("")

g_exportedScripts = message_button, clear_button,
```

Status | Compiler

```
10:33:34: This is Geany 1.36.
10:33:34: File /home/ian/vte_code_for_alpha_instruction.py opened (1).
10:33:34: File /home/ian/dev/alpha/alpha_instruction_builder_4.py opened (2).
10:33:34: File /home/ian/dev/alpha/all_instructions_1.py opened (3).
10:33:34: File /home/ian/dev/alpha/treeview/treview_example_1.py opened (4).
```

line: 1 / 16 col: 0 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: unknown

APSO ~ Dependency Notes

- Advised to install the MRI extension for LibreOffice
- Install the default-jre.

```
$ sudo apt install jre-default
```

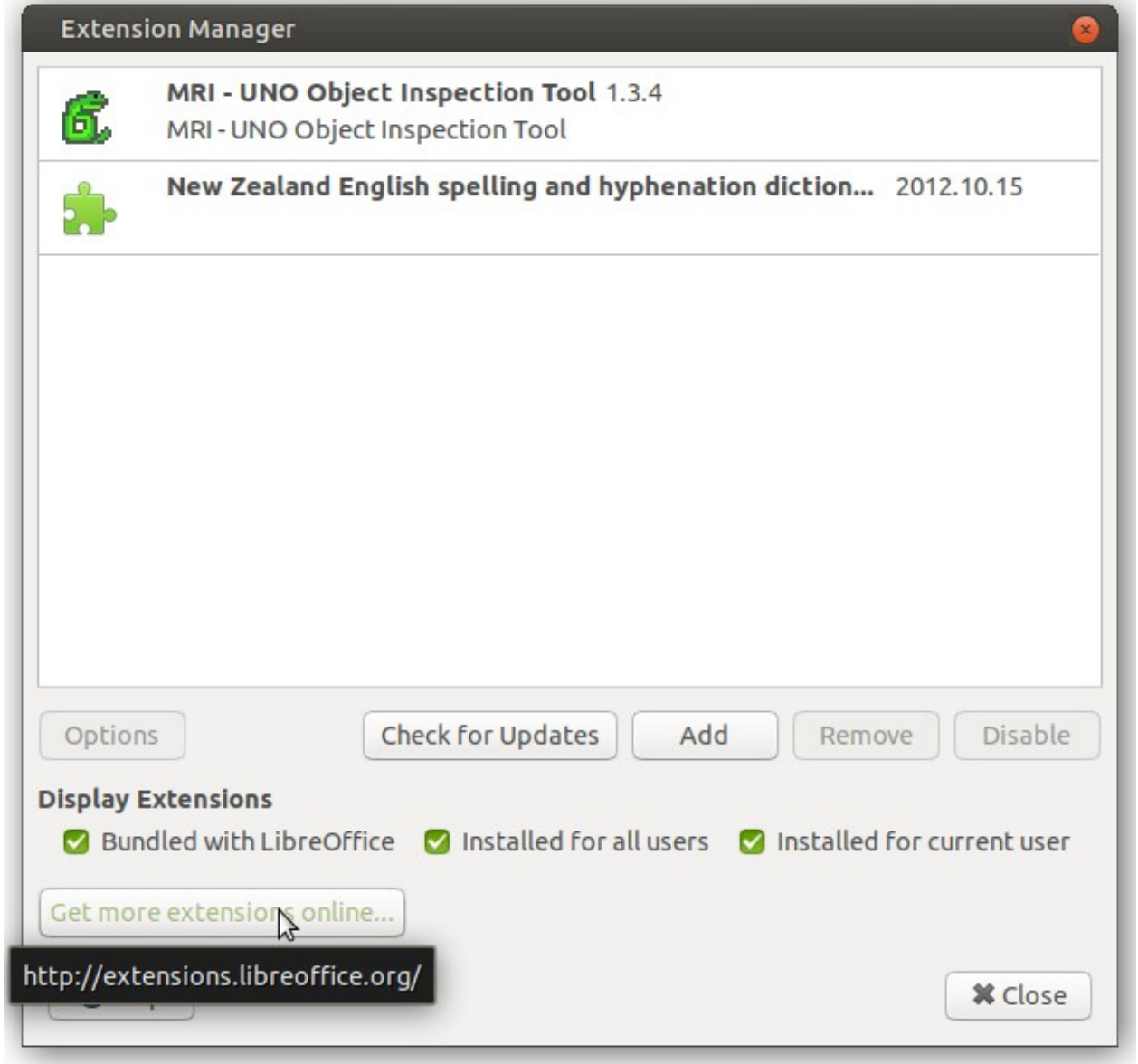
- Install libreoffice-script-provider-python

```
$ sudo apt install libreoffice-script-provider-python
```

APSO ~ Download and Addon of Extension

APSO

Performing Addon Extension for LibreOffice



APSO Addon

extensions.libreoffice.org/?q=APSO

Extensions

LibreOffice Extensions, Documentation and Templates repository

APSO

Extensions About Login

What are you looking for?

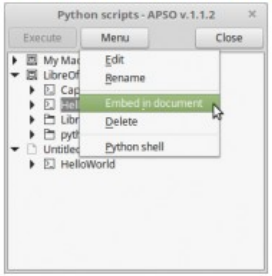
APSO

SEARCH

Add tag filters: **Base (12)** Business (106) **Calc (134)** Color Palette (14) Database (12) Dictionary (108) Documentation (34) Documents (169) **Draw (32)** Drawings (27) Education (100) Extensions (209) Fun (47) Gallery (32) Icons (6) **Impress (57)** Macro (13) **Math (5)** PDF (1) Presentations (58) Spreadsheets (146) Templates (505) **Writer (197)**

Sort order:

APSO - Alternative Script Organizer for Python



Based on an original script from Hanya, APSO will install a macro organizer dedicated to python scripts.

06-2021 ★★☆☆☆

Extensions → Extensions / About / Login

Based on Theme by Sara (Innovatif) / Powered by SilverStripe / Impressum (Legal Info) / Privacy Policy (Datenschutzerklärung)

APSO Addon.

Extensions

LibreOffice Extensions, Documentation and Templates repository

search

Extensions About Login

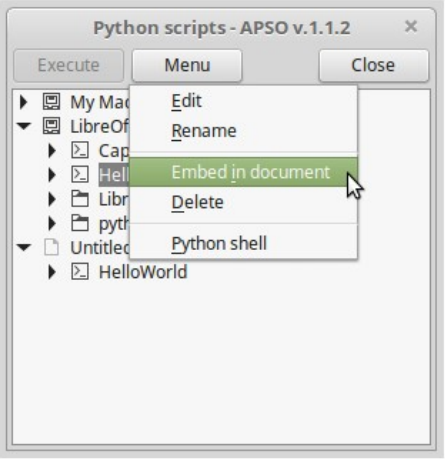
German | English

APSO - Alternative Script Organizer for Python

Tags: *Extensions*

Rating: ★★☆☆☆

Based on an original script from Hanya, APSO will install a macro organizer dedicated to python scripts.



Description

APSO is compatible with OpenOffice

Based on an original script from Hanya, APSO is an extension that will install a macro organizer dedicated to python scripts.

Prerequisite

APSO Addon.
Download to Downloads folder...

APSO Options

The extension option page is available under *Tools -> Extension Manager -> APSO -> Options*.
By default, APSO uses the system defined editor. To edit scripts with a specific editor, provide the executable full path in the "Editor" field.
To allow the opening at a given line and column offset when relevant, enter in the "Options" field the command line syntax corresponding to the choosen editor, using the placeholders {FILENAME}, {ROW} and {COL} (they will be replaced in due time).
Example for Emacs: +{ROW}:{COL} {FILENAME}.
Example for Sublime Text: {FILENAME}:{ROW}:{COL}.

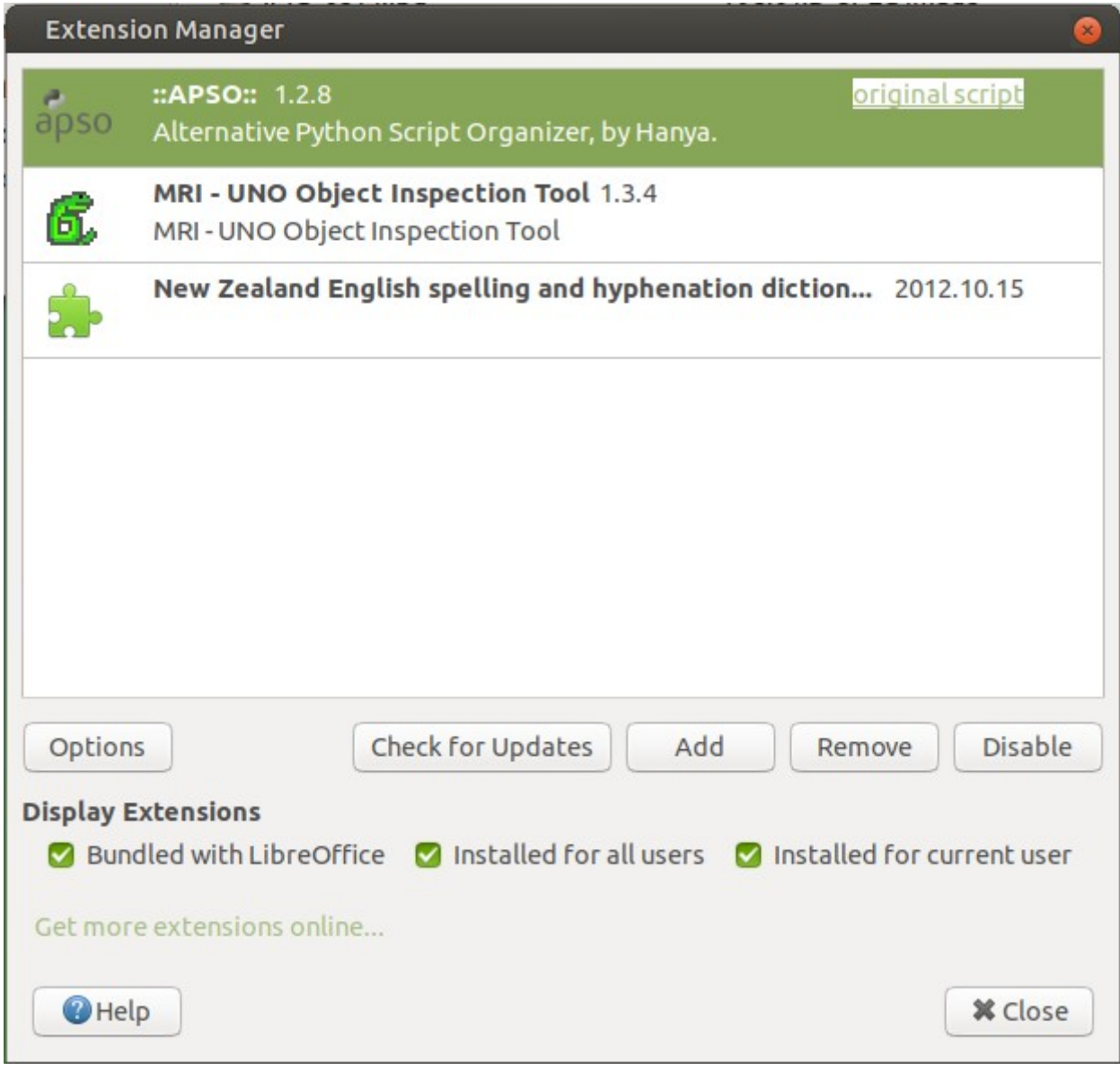
Homepage: https://gitlab.com/jmzambon/apso

Release List

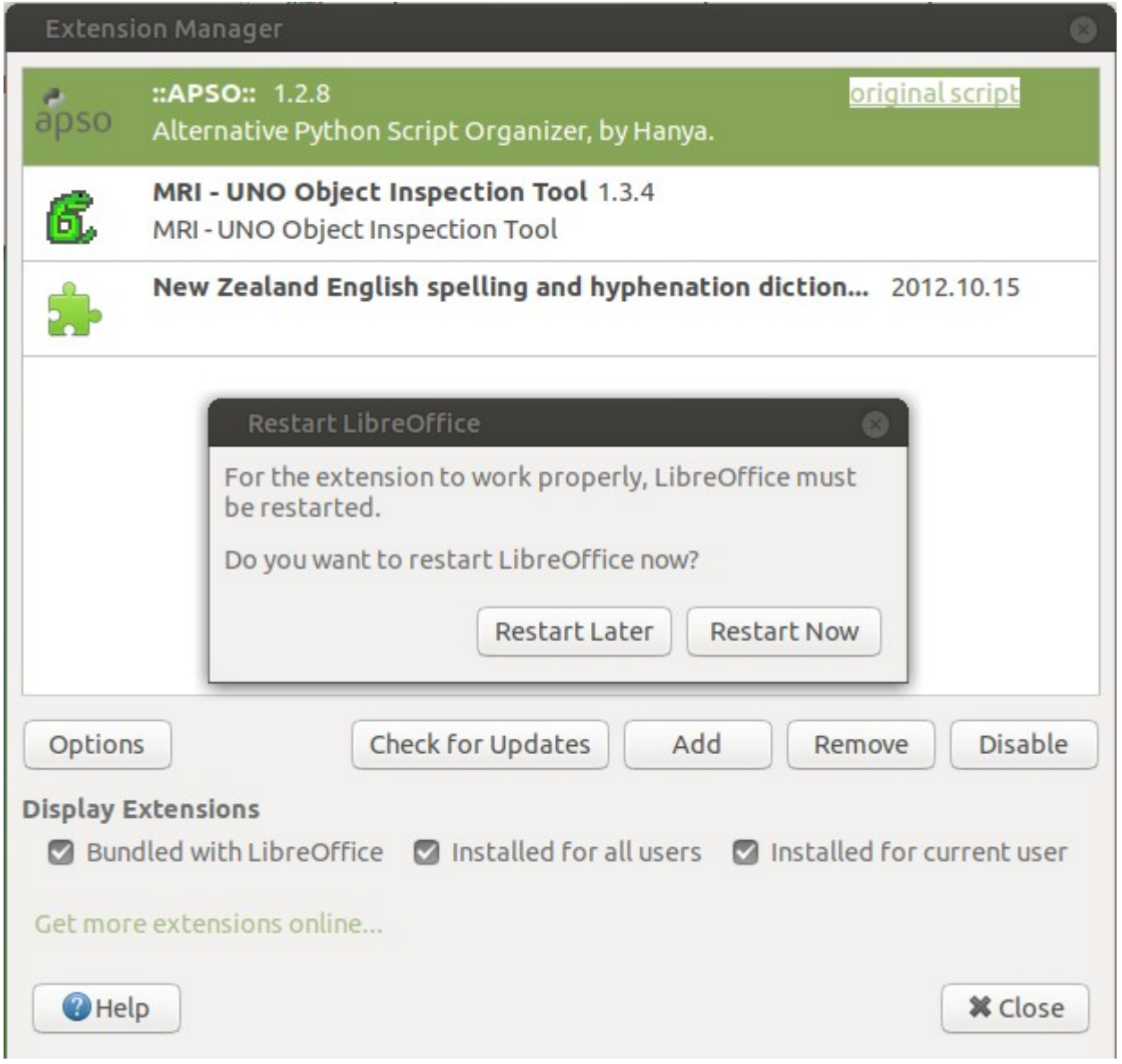
Release	Description	Compatibility	Operating Systems	License	Release notes	
v1.2.8	Bug fix and small enhancement.	3.4	Linux, Windows, macOS	AL	- The console() method now retuns a reference to the console object, allowing to close programmaticaly the console window by invoking its enddialog() method (issue #21). - Fix issue #20 "LibreOffice freezes on some linux configurations".	DOWNLOAD
1.2.7	Minor improvements	3.4	Linux, Windows, macOS	AL	Make node sorting locale aware. Minor improvements and bug fixes	DOWNLOAD
			Linux,		The default header lines, inserted at creation of a new module, can now be customized from the option dialog.	

APSO Install addon from Downloads folder

APSO addon installed.

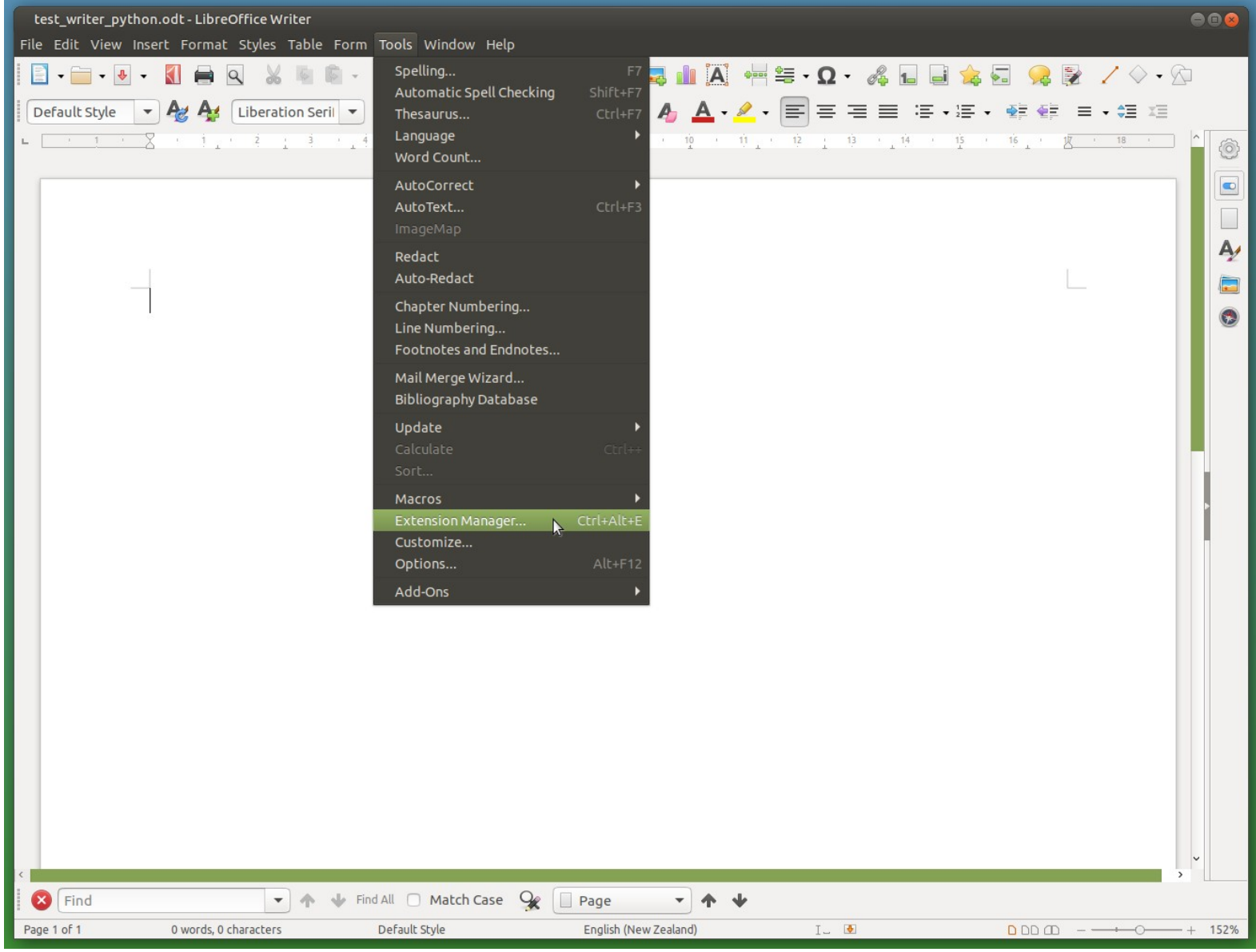


APSO addon installed.

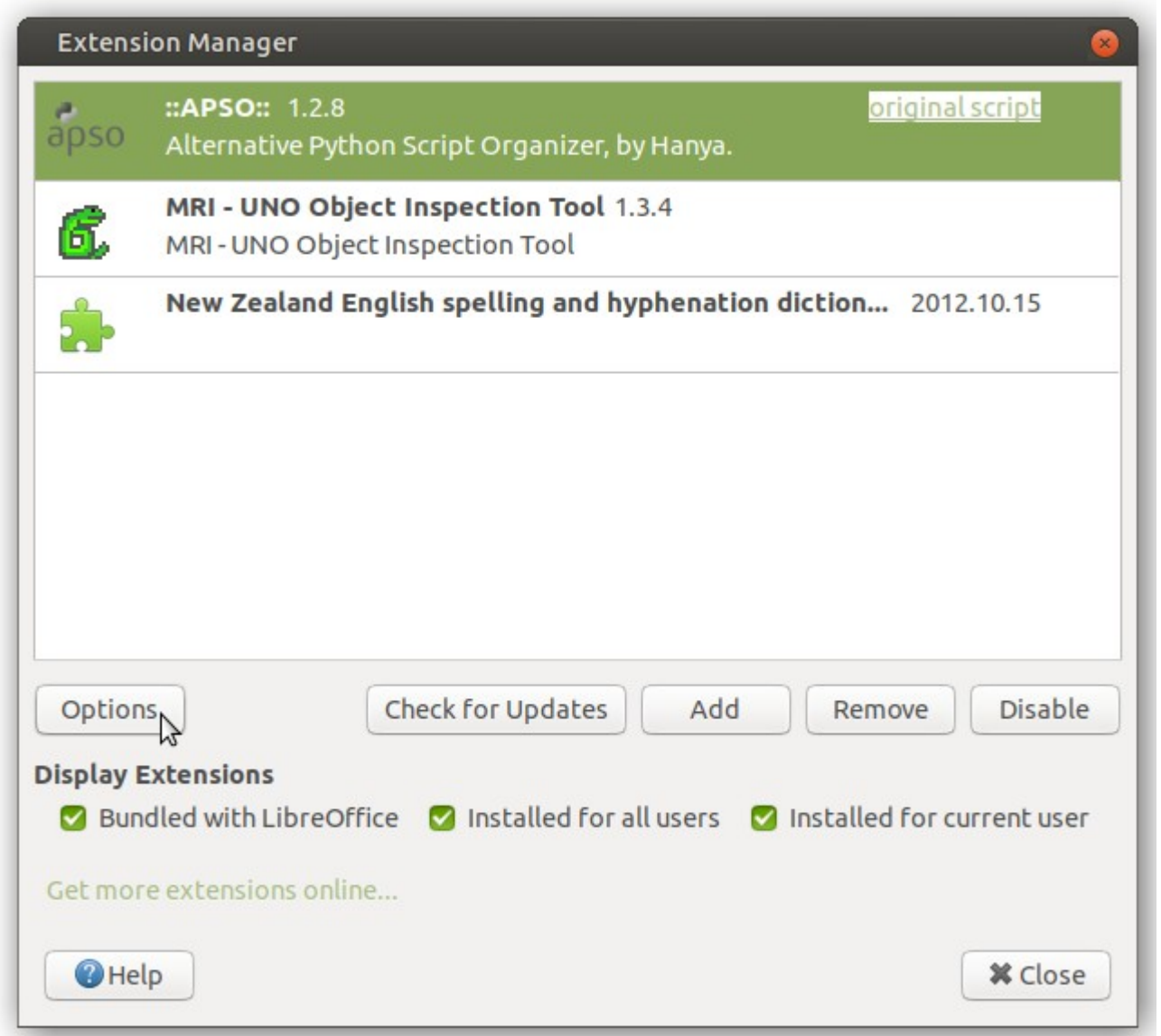


APSO
Tailoring editor.

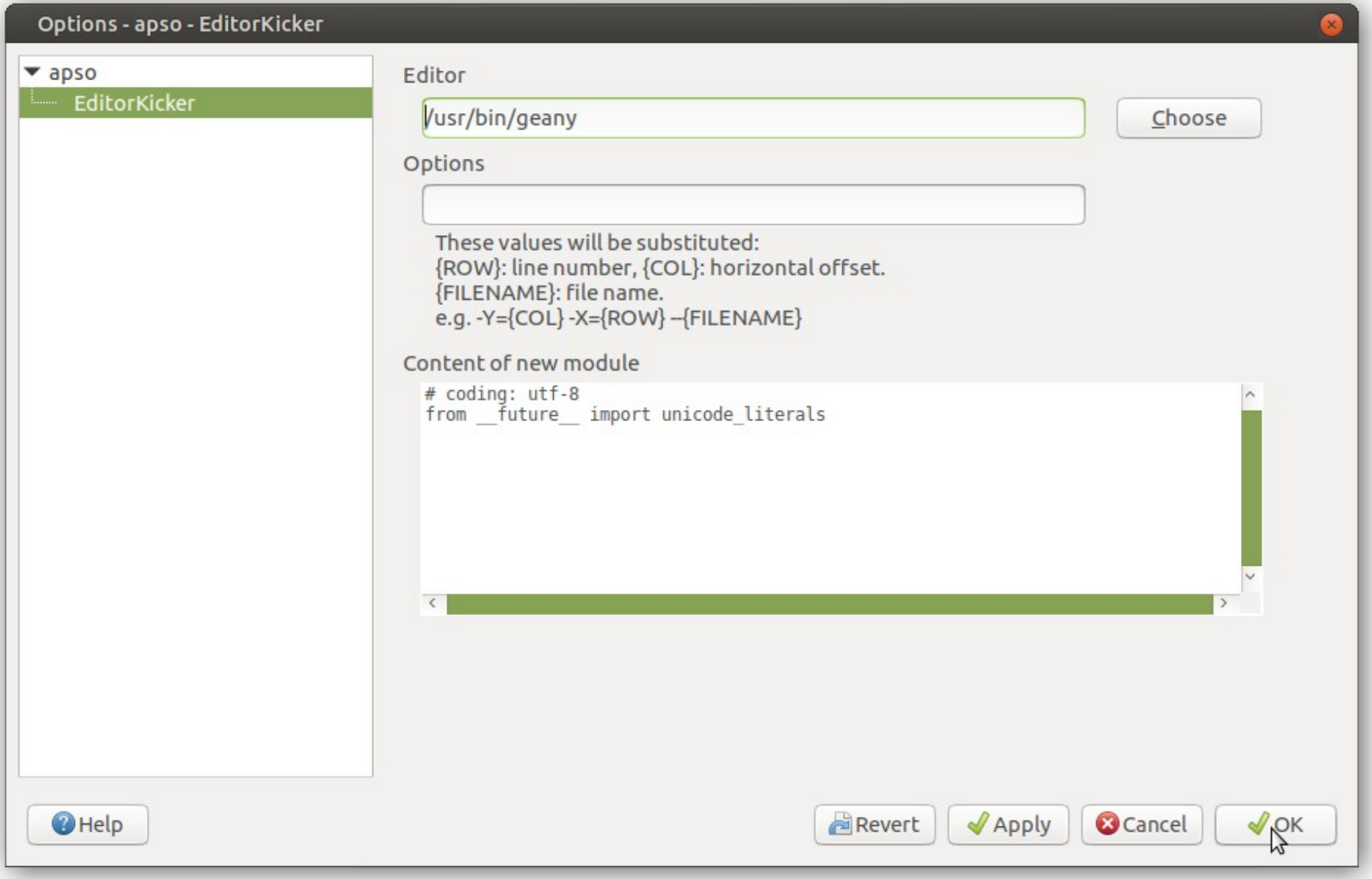
Tools -->
Extension Manager...



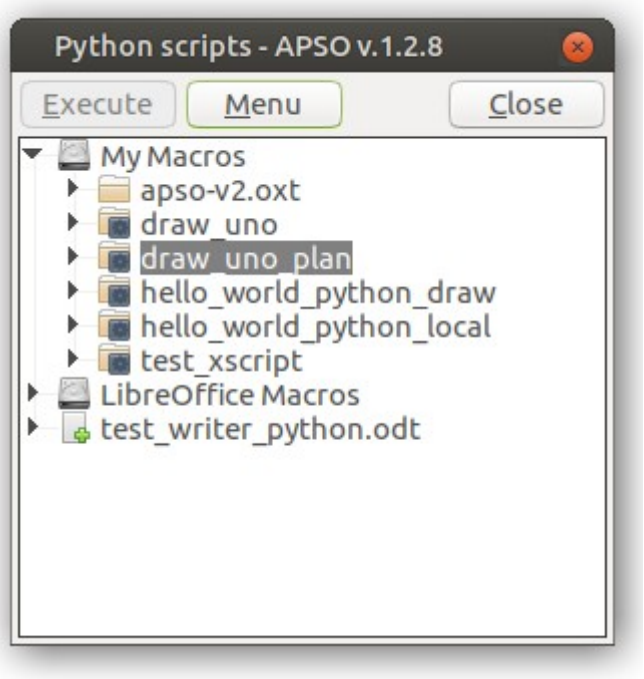
Tools -->
Extension Manager... -->
APSO --> Options

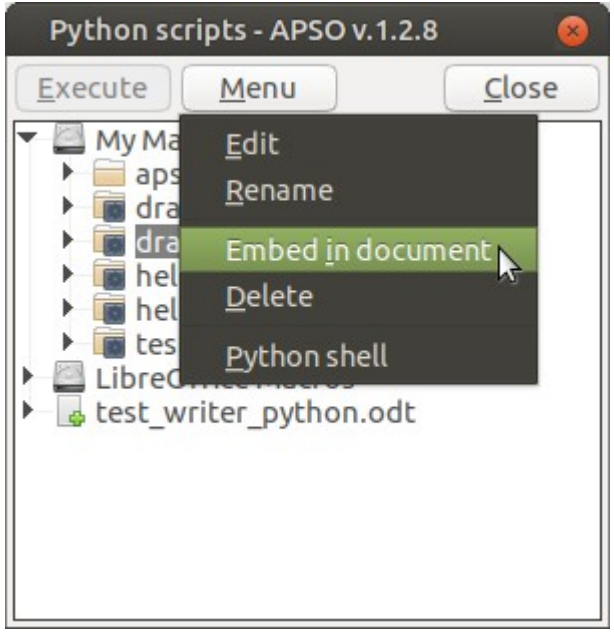


APSO. Choose Python IDE...

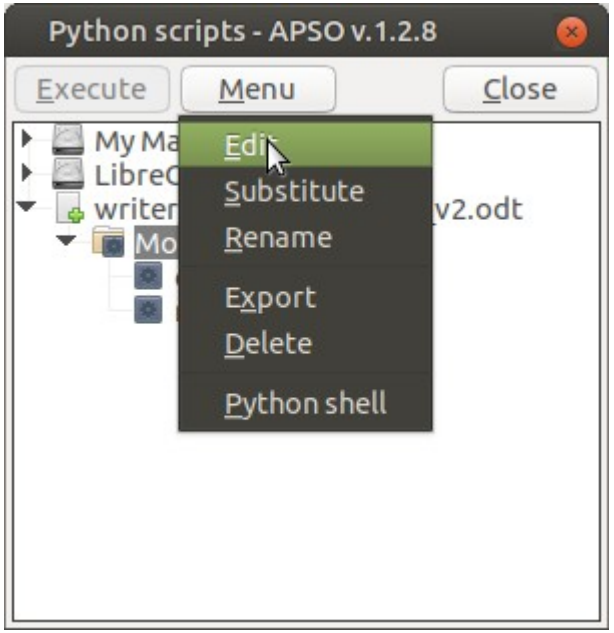


Python Moving from My Macros to Embedded in document.





Python Embedded in document. Start Editing.



Python Embedded in document. Editing using Geany.

Module.py - /tmp/lu2479jkpekt.tmp/1/Scripts/python - Geany

File Edit Search View Document Project Build Tools Help

Symbols | Documents

~
- get_alpha_sys_c_data.py
- vte_code_for_alpha_instruction.py

~/dev/alpha
- all_instructions_1.py
- alpha_instruction_builder_4.py
- get_alpha_cpu_data.py
- get_description.py
- get_instructions_no_r31.py
- hello_css.py
- image_embedding_...pixbuf_headerbar.py
- status_bar.py
- test_table_in_sys_c_data.py

~/dev/alpha/alpha_in...ction_builder/opcodes
- alpha_gui_geany_4.py

~/dev/alpha/treeview
- treview_example_1.py

~/dev/builder
- double_precision.py

/tmp/lu2479jkpekt.tmp/1/Scripts/python
- Module.py

get_instructions_no_r31.py | alpha_gui_geany_4.py | Module.py

```python
#!/usr/bin/env python3
import uno, sys, os, zipimport

def get_python_script(button):
    url = button.value.Source.Model.Parent.Parent.Parent.URL
    url = url.split("//")[1] + "/Scripts/python/"
    importer = zipimport.zipimporter(url)
    return importer.get_source('Module')

def message_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    s = "Hello World! ~ Using Python: " + sys.version.split(" ")[0]
    doc.getText().setString(s + "\n\n" + get_python_script(button))

def clear_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("")

g_exportedScripts = message_button, clear_button,
```

Status | Compiler

```
16:53:43: This is Geany 1.36.
16:53:43: File /home/ian/vte_code_for_alpha_instruction.py opened (1).
16:53:43: File /home/ian/dev/alpha/alpha_instruction_builder_4.py opened (2).
16:53:43: File /home/ian/dev/alpha/all_instructions_1.py opened (3).
16:53:43: File /home/ian/dev/alpha/treeview/treview_example_1.py opened (4).
```

This is Geany 1.36.

XSCRIPTCONTEXT

XSCRIPTCONTEXT.

http://www.openoffice.org/api/docs/common/ref/com/sun/star/script/provider/XScriptContext.html

:: com :: sun :: star :: script :: provider ::

unpublished

interface XScriptContext

Usage Restrictions

not published

Description

This interface is provided to scripts, and provides a means of access to the various interfaces which they might need to perform some action on a document. It is required to be passed as the first argument for any Java scripts.

Methods' Summary	
getDocument	Obtain the document reference on which the script can operate
getInvocationContext	provides access to the context where the script was invoked
getDesktop	Obtain the desktop reference on which the script can operate
getComponentContext	Obtain the component context which the script can use to create other uno components

PyUNO Samples
https://wiki.openoffice.org/wiki/PyUNO_samples

Hello World.py

This script print a Hello World in a Writer document. This uses the XText interface, the actual content is on the String value of the com.sun.star.text.XTextRange .

First step we define the HelloWorldPython() class, then we call the XSCRIPTCONTEXT which will let us work within a sub-environment from UNO. Notice we didn't import uno and we don't need to connect to Apache OpenOffice through the socket. Also we have the uno environment by calling getDocument().

Then we call the XText interface to call on the END and STRING methods to write up the "Hello World (in Python)".

```
# HelloWorld python script for the scripting framework

def HelloWorldPython( ):
    """Prints the string 'Hello World(in Python)' into the current document"""
#get the doc from the scripting context which is made available to all scripts
    model = XSCRIPTCONTEXT.getDocument()
#get the XText interface
    text = model.Text
#create an XTextRange at the end of the document
    tRange = text.End
#and set the string
    tRange.String = "Hello World (in Python)"
    return None
```

Python code resides in “My Macros & Dialogs”:
`~/.config/libreoffice/4/user/Scripts/python/TableSample.py`
OR LibreOffice Macros & Dialogs:
`/usr/lib/libreoffice/share/Scripts/python/pythonSamples/TableSample.py`
Create a socket:

```
$ libreoffice --writer -accept="socket,host=localhost,port=2002;urp;StarOffice.ServiceManager"
```

TableSample.py

On this script we abandon the XSCRIPTCONTEXT and use the UNO module. We also use nested modules from Text, AWT, and LANG. Since we using the UNO module we have to import what we need straight from the com.sun.star path.

The first step is to get the functions for generating the document, and the other for manipulating the content. The document and content is under the **createTable()**, while **insertTextIntoCell()** will handle the positioning within the table.

```
import uno

# a UNO struct later needed to create a document
from com.sun.star.text.ControlCharacter import PARAGRAPH_BREAK
from com.sun.star.text.TextContentAnchorType import AS_CHARACTER
from com.sun.star.awt import Size

from com.sun.star.lang import XMain
```

Instead of:
`desktop = XSCRIPTCONTEXT.getDesktop()`

```
def createTable():
    """creates a new writer document and inserts a table with
    ctx = uno.getComponentContext()
    smgr = ctx.ServiceManager
    desktop = smgr.createInstanceWithContext( "com.sun.star.frame.Desktop",ctx)

    # open a writer document
    doc = desktop.loadComponentFromURL( "private:factory/swriter","_blank", 0, () )
```

...more codes... See: https://wiki.openoffice.org/wiki/PyUNO_samples

BASIC Sample. Retrieve and Display

writer_basic_example.odt - LibreOffice Writer

File Edit View Insert Format Styles Table Form Tools Window Help

Message Clear

```
Hello World! ~ Using BASIC

REM  *****  BASIC  *****

function Main()as string
      ' Get the BASIC code
      if BasicLibraries.getByName("Standard").hasByName("Module1") then
            Main = BasicLibraries.getByName("Standard").getByName("Module1")
      end if
end function

sub message_button(button)
      doc = ThisComponent
      doc.getText().setString("Hello World! ~ Using BASIC" + _
              chr(13) + chr(13) + Main())
end sub

sub clear_button(button)
      doc = ThisComponent
      doc.getText().setString("")
end sub
```

Python Sample. Retrieve and Display

writer_python_example.odt - LibreOffice Writer

File Edit View Insert Format Styles Table Form Tools Window Help

Default Style | DejaVu Sans M | 10

Message | Clear

```
Hello World! ~ Using Python: 3.8.10

#!/usr/bin/env python3
import uno, sys, os, zipimport

def get_python_script(button):
    url = button.value.Source.Model.Parent.Parent.Parent.URL
    url = url.split("//")[1] + "/Scripts/python/"
    importer = zipimport.zipimporter(url)
    return importer.get_source('Module')

def message_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    s = "Hello World! ~ Using Python: " +  sys.version.split(" ")[0]
    doc.getText().setString(s + "\n\n" + get_python_script(button))

def clear_button(button):
    doc = XSCRIPTCONTEXT.getDocument()
    doc.getText().setString("")

g_exportedScripts = message_button, clear_button,
```

Demo:

Floor Plan ~ Draw document
draw_embedded_python_plan.odg

Amortization ~ Calc document
calc_embedded_python_amortization.ods

APSO ~~Megbox~~ Msgbox
writer_python_msgbox.odt

End.