

Getting Closer to a Help Language

Olivier Hallot
The Document Foundation
olivier.hallot@libreoffice.org

Why a Help language?

- Historically, developers don't like documentation
- Thus, the gap between features and Help is wide(-ning)
 - Speed of development is faster than documentation
- Coupling of Help pages and UI elements is a challenge
- Focus on user experience

Should the Help language be...

- Open
 - “ça va sans dire”
- Flexible
 - Allows specific situations, translations
- Precise
 - Plays as reference for regressions/feature description
- Contributor-friendly
 - Our challenge every day
- Keep the pace of development

Specifics of LibreOffice Help

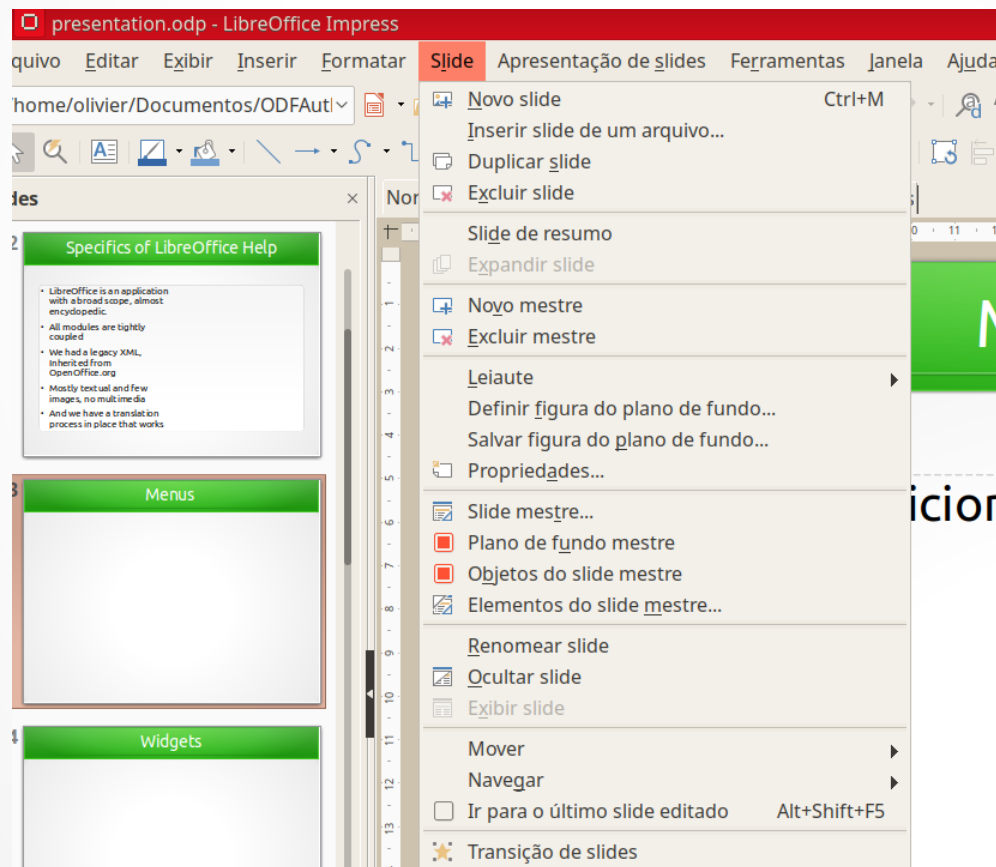
- LibreOffice is an application with a broad scope, almost encyclopedic.
- All modules are tightly coupled
- We had a legacy XML, inherited from OpenOffice.org
- Mostly textual and few images, no multimedia
- And we have a translation process in place that works

Proposal of a Help language?

- A set of XML and namespaces capable of describing
 - Menu paths
 - Widgets
 - Screenshots & images
 - Icons & toolbars
 - Multimedia
 - Guide the user
 - Indexing & searches
 - Save translation work

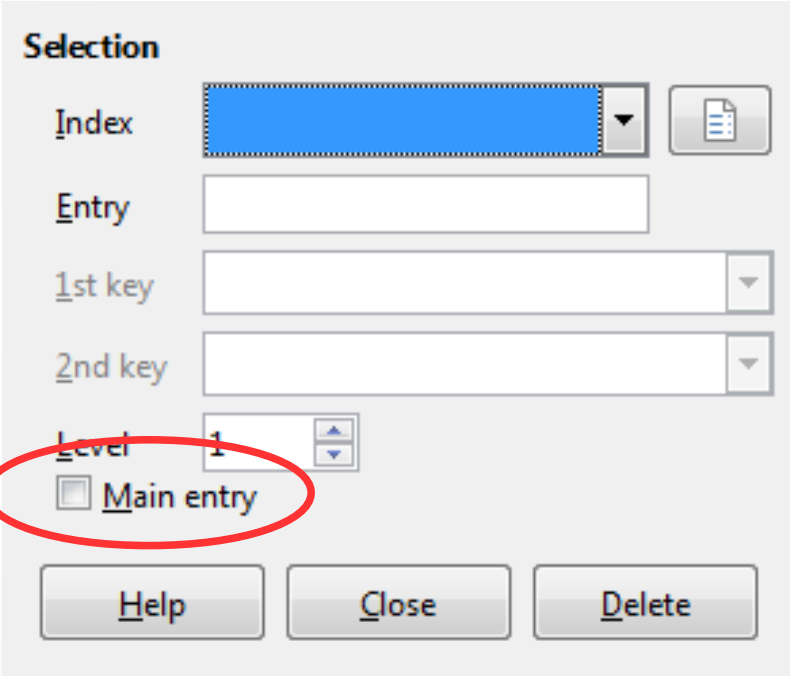
Menus

- Describe the menu paths for the command
- Use the menu XML + namespace to transform in a textual path



Widgets

- Mark precisely the content and shape of dialog widgets
- Use the dialog XML + namespace to transform in a visual reference



The image shows a dialog box titled "Selection". It contains the following elements:

- Index**: A blue selection box with a dotted border and a dropdown arrow, next to a document icon button.
- Entry**: A text input field.
- 1st key**: A text input field with a dropdown arrow.
- 2nd key**: A text input field with a dropdown arrow.
- Level**: A spinner control showing the value "1".
- Main entry**: A checkbox, which is circled in red.
- Buttons**: "Help", "Close", and "Delete" buttons at the bottom.

Icons and toolbars

- Describe icons and toolbars
- Follow improvements in icon designs
- Use toolbars XML + namespace

Screenshots

- Today, screenshots are semi-automated
- Depends on systems UI
- Use dialog XML to generate image

Sort Criteria

	Column	Key type	Order
<input checked="" type="checkbox"/> Key 1	1		<input type="radio"/> Ascending <input checked="" type="radio"/> Descending
<input type="checkbox"/> Key 2	1		<input type="radio"/> Ascending <input checked="" type="radio"/> Descending
<input type="checkbox"/> Key 3	1		<input type="radio"/> Ascending <input checked="" type="radio"/> Descending

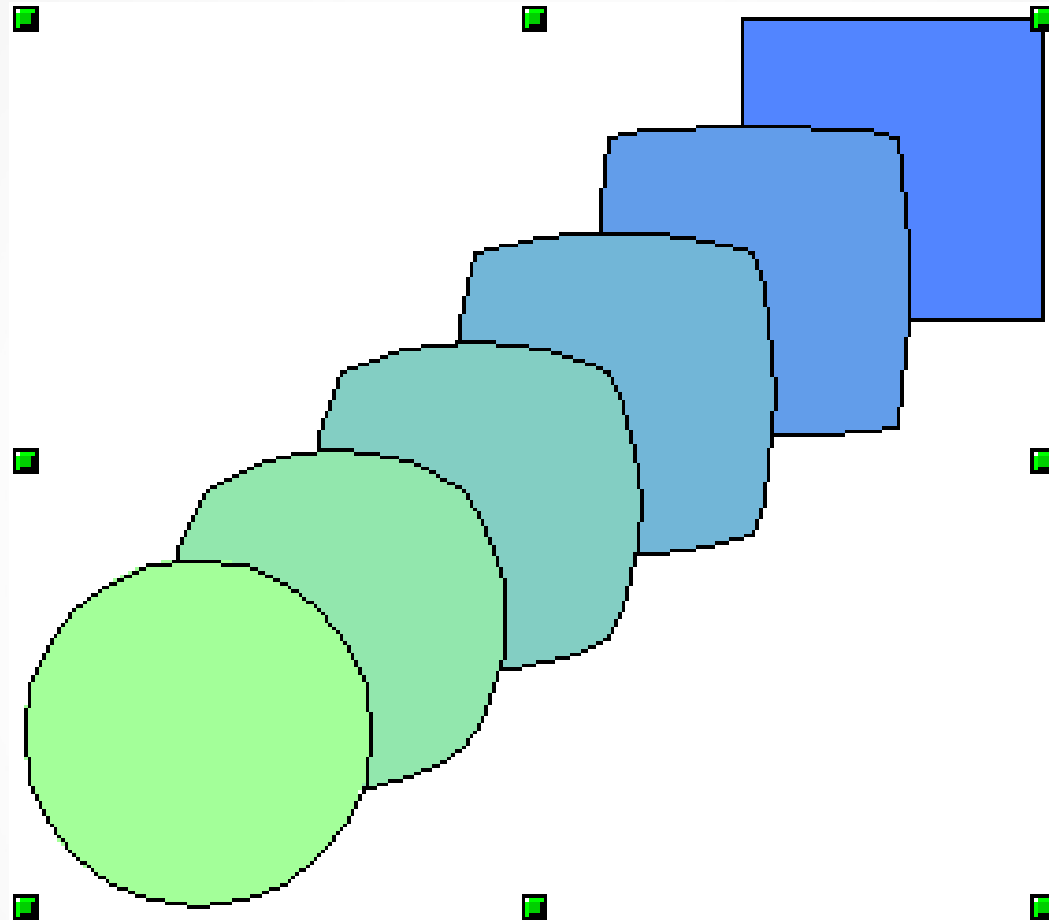
Direction
 Columns
 Rows

Separator
 Tabs
 Character

Language

Setting
 Match case

Screenshots & Images

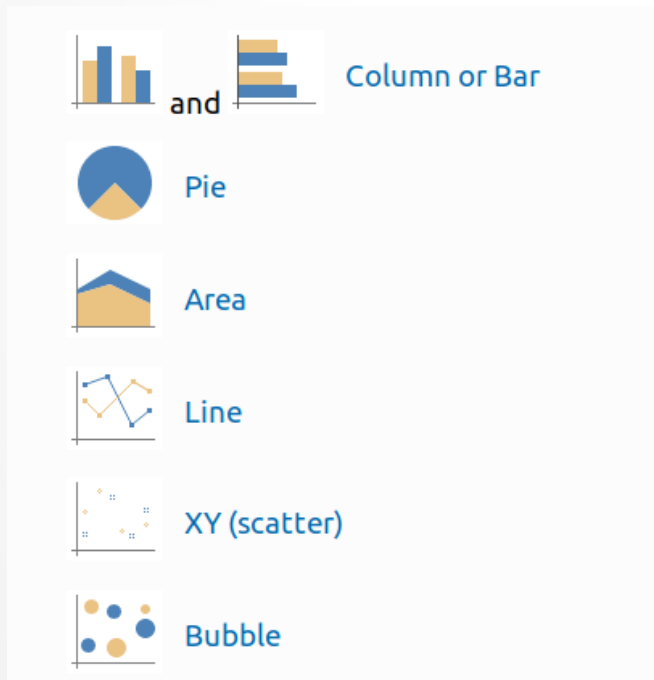


Context dependent switches

- Context switches
 - System (MAC, WIN, UNIX)
 - Application (WRITER, CALC, BASE...)
- Inline context switches
- Block context switches

Block-class

- Code, pycode, bascode
- Specialized tables
 - Icon tables



screen_io Python module

```
1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 def MsgBox(prompt: str, buttons=0,
5           """ Displays a dialogue box co
6           xScript = _getScript("_MsgBox"
7           res = xScript.invoke((prompt,b
8           return res[0]
9
10 def InputBox(prompt: str, title='L
11 """ Displays a dialogue box co
```

Paragraph-class

- Notes, Tips and Warnings
- Headings



O ícone **Importante!** fornece informações importantes sobre a segurança dos dados e do sistema.



O ícone **Nota** indica informações extras: por exemplo, maneiras alternativas de alcançar uma dada meta.



O ícone **Dica** fornece dicas sobre como trabalhar com o programa de maneira mais eficaz.

Characters-class

- Literal
- Input
- Widget
- MenuItem
- keycode

We created an online Editor

LibreOffice Documentation XHP Editor

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <helpdocument version="1.0">
3   <!--
4     * This file is part of the LibreOffice project.
5     *
6     * This Source Code Form is subject to the terms of the Mozilla Public
7     * License, v. 2.0. If a copy of the MPL was not distributed with this
8     * file, You can obtain one at http://mozilla.org/MPL/2.0/.
9     *
10    -->
11   <meta>
12     <topic id="text/sbasic/python/Python_Programming">
13       <title id="tit" xml-lang="en-US">Python : Programming with Python</title>
14       <filename>/text/sbasic/python/python_programming.xhp</filename>
15     </topic>
16   </meta>
17   <body>
18     <bookmark branch="index" id="N0218">
19       <bookmark_value>Python;Programming</bookmark_value>
20       <bookmark_value>XSCRIPTCONTEXT;Python</bookmark_value>
21       <bookmark_value>uno.py</bookmark_value>
22     </bookmark>
23   </body>
24 </helpdocument>
```

Actions:

File name:

File name:

Edit:

Document:

Bookmarks:

Sections:

Tables:

Paragraph:

Characters:

Headings:

Switches:

Lists:

Links:

Rendered page

Programming with Python Scripts


A Python macro is a function within a .py file, identified as a module. Unlike LibreOffice Basic and its dozen of [UNO objects functions or services](#), Python macros use the XSCRIPTCONTEXT UNO single object, shared with JavaScript and Beanshell. The `g_exportedScripts` global tuple explicitly lists selectable macros from a module. Python modules hold autonomous code logic, and are independent from one another.

XSCRIPTCONTEXT Global Variable


Genuine Basic UNO facilities can be inferred from XSCRIPTCONTEXT global variable. Refer to LibreOffice API for a complete [description of XSCRIPTCONTEXT](#). XSCRIPTCONTEXT methods summarize as:

Methods	Description	Mapped in Basic as
<code>getDocument()</code>	The document reference on which the script can operate.	ThisComponent
<code>getDesktop()</code>	The desktop reference on which the script can operate.	StarDesktop
<code>GetComponentContext()</code>	The component context which the script can use to create other uno components.	GetDefaultContext

HelloWorld and **Capitalise** installation shared scripts illustrate UNO-related macros making use of XSCRIPTCONTEXT global variable.

 Python standard output file is not available when running Python macros from **Tools - Macros - Run Macro** menu. Refer to [Input/Output to Screen](#) for more information.

Module import

 XSCRIPTCONTEXT is not provided to imported modules.

LibreOffice Basic libraries contain classes, routines and variables, Python modules contain classes, functions and variables. Common pieces of reusable Python or UNO features must be stored in [My macros](#) within `<User Profile>/Scripts/python/pythonpath`. Python libraries help organize modules in order to prevent module name collisions. Import `uno.py` inside shared modules.

Genuine BASIC UNO facilities can be inferred using `uno.py` module. Use [Python interactive shell](#) to get a complete module description using `dir()` and `help()` Python commands.

What's next ?

- It is our interest to investigate XML + namespace conversion of
 - Dialogs
 - Menus
 - toolbars
- We also would like to converge our Help with existing OpenDocument Guides (Convergence)
- We're open to suggestions and collaboration
- Join our community!

Thank you !