

# LibreOffice Development Workshop

## LibreOffice Automation with Basic and Python Scripts

Rafael Henrique Palma Lima

*UTFPR / TDF Member*

Sponsors:



Collabora



allotropia



# Agenda

This session is divided in three topics:

- LibreOffice automation using Basic (Rafael Lima)
- LibreOffice automation using Python (Alain Romedenne)
- Using the ScriptForge library in Basic and Python (Jean-Pierre Ledure)



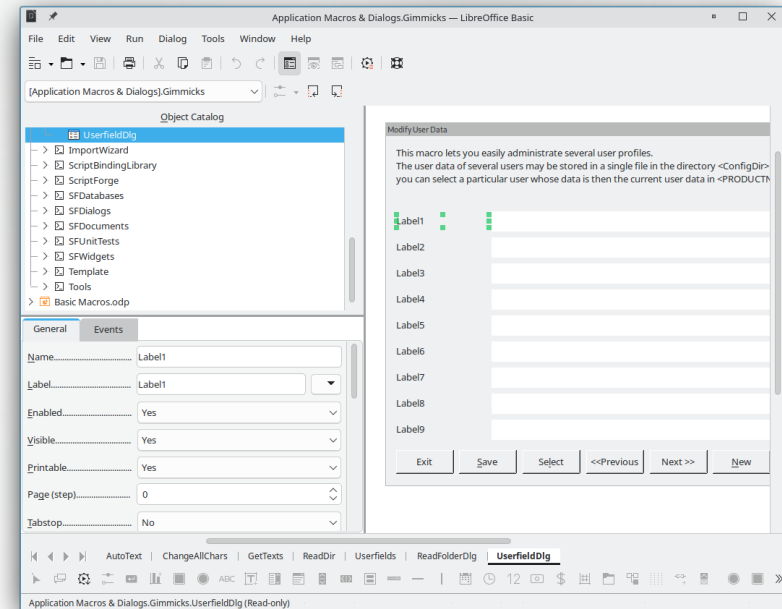
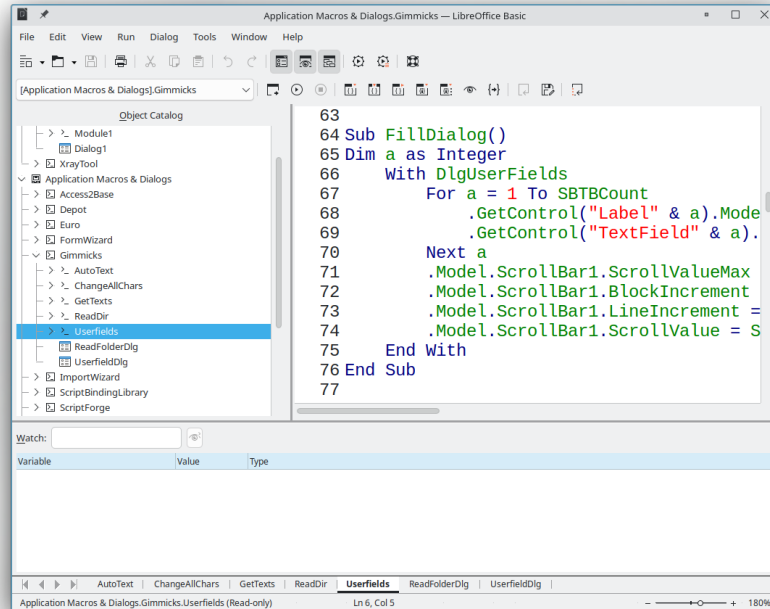
# Why Scripting?

- Automate common and repetitive tasks
- Extend application functionalities (macros and extensions)
- Interact directly with the LibreOffice API
- Create custom applications tailored for your needs (integrate databases, create dialogs, forms, etc)



# Scripting with Basic

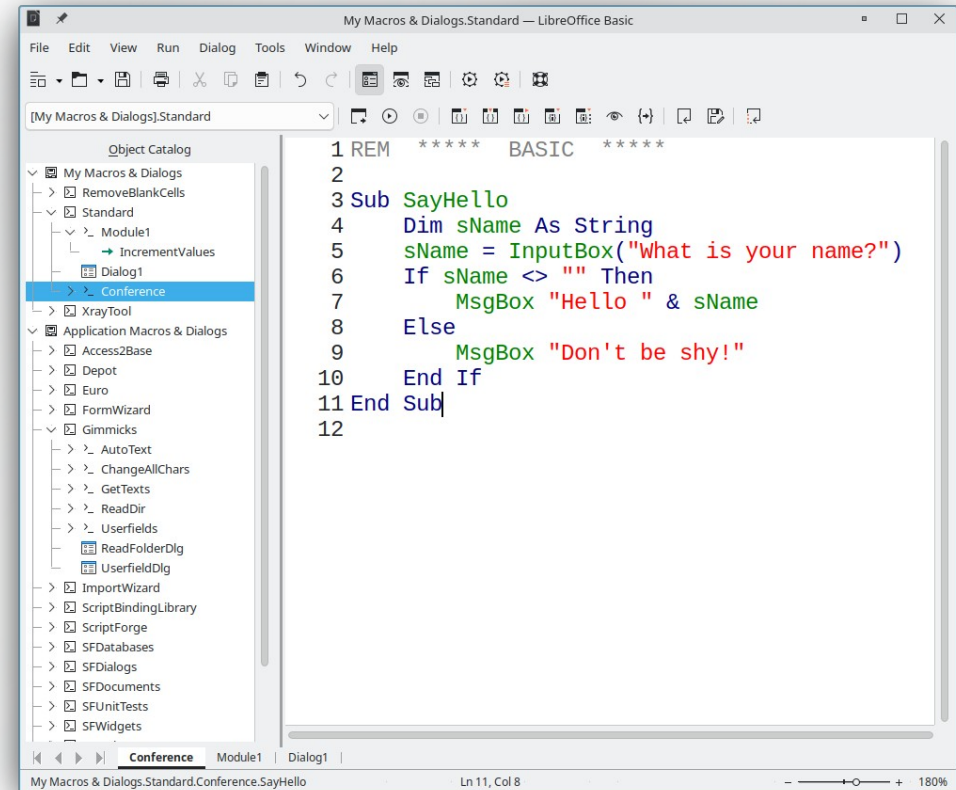
- LibreOffice has a built-in IDE for creating scripts with the Basic language
- The Dialog Editor can be used to create and edit dialogs





## A “Hello World” script:

- 1) Open any LibreOffice application (f.i. open Calc)
- 2) Go to Tools – Macros – Edit Macros
- 3) A default module named “Module1” from the “Standard” library will be selected
- 4) Create the Sub SayHello by writing the next code
- 5) Place the cursor anywhere inside the newly created Sub
- 6) Click the Run button (or press F5)



The screenshot shows the LibreOffice Basic editor window titled "My Macros & Dialogs.Standard — LibreOffice Basic". The left pane displays the "Object Catalog" with a tree structure. Under "My Macros & Dialogs", "Standard" is expanded, showing "Module1" (selected) and "Dialog1". Under "Application Macros & Dialogs", "XrayTool" is expanded, showing "Conference" (selected). The right pane shows the BASIC code for the "SayHello" subprocedure:

```
1 REM ***** BASIC *****
2
3 Sub SayHello
4   Dim sName As String
5   sName = InputBox("What is your name?")
6   If sName <> "" Then
7       MsgBox "Hello " & sName
8   Else
9       MsgBox "Don't be shy!"
10  End If
11 End Sub
12
```

The status bar at the bottom indicates "My Macros & Dialogs.Standard.Conference.SayHello" and "Ln 11, Col 8".

**A more complex script:** The following script creates a Writer document and inserts some text into it

```
Sub CreateWriterDoc
    Dim sName As String
    sName = InputBox("What is your name?")
    ' Only create the doc if a name is given
    If sName <> "" Then
        Dim oDoc As Object
        Dim oText As Object
        Dim sNewText As String
        oDoc = StarDesktop.loadComponentFromUrl("private:factory/swriter", _
                                                "_blank", 0, Array())

        oText = oDoc.getText()
        sNewText = "Hello " & sName & Chr(10)
        sNewText = sNewText & "Today is " & Date()
        oText.insertString(oText.End, sNewText, False)
    End If
End Sub
```

# How do I learn this?

StarDesktop: is a runtime object that gives access to the entire LibreOffice application

[https://api.libreoffice.org/docs/idl/ref/servicecom\\_1\\_1sun\\_1\\_1star\\_1\\_1frame\\_1\\_1Desktop.html](https://api.libreoffice.org/docs/idl/ref/servicecom_1_1sun_1_1star_1_1frame_1_1Desktop.html)

[https://wiki.documentfoundation.org/Documentation/DevGuide/LibreOffice\\_Basic#Accessing\\_the\\_UNO\\_API](https://wiki.documentfoundation.org/Documentation/DevGuide/LibreOffice_Basic#Accessing_the_UNO_API)

LoadComponentFromURL

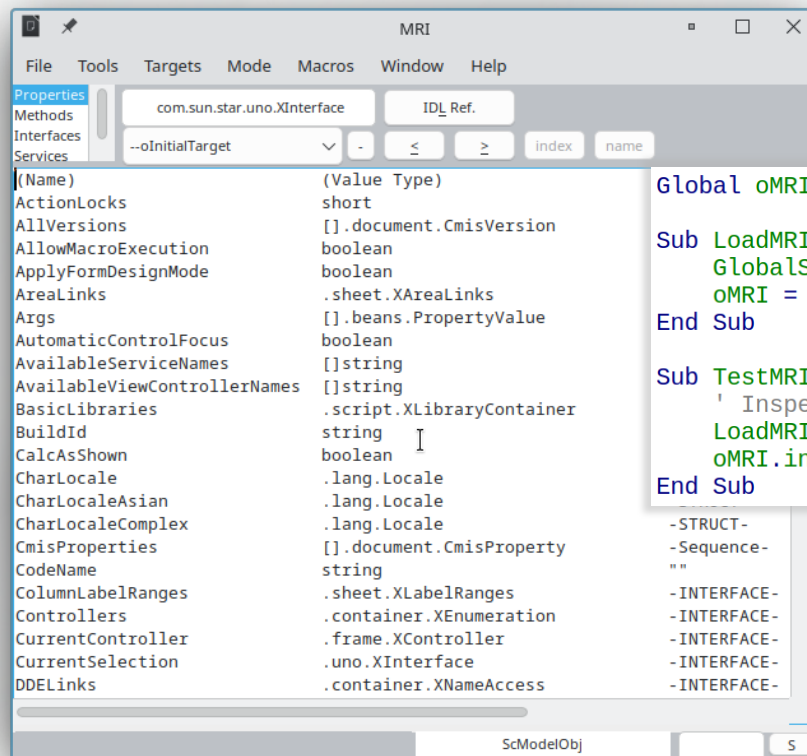
[https://wiki.documentfoundation.org/Documentation/DevGuide/Text\\_Documents#Creating\\_and>Loading\\_Text\\_Documents](https://wiki.documentfoundation.org/Documentation/DevGuide/Text_Documents#Creating_and>Loading_Text_Documents)

# Inspecting Objects

Use the MRI or Xray extensions to inspect objects and discover its properties and methods

Download Link:

<https://github.com/hanya/MRI>



Global oMRI As Object

```
Sub LoadMRI
    GlobalScope.BasicLibraries.LoadLibrary("MRILib")
    oMRI = CreateUnoService("mytools.Mri")
End Sub
```

```
Sub TestMRI
    ' Inspect current document
    LoadMRI
    oMRI.inspect(ThisComponent)
End Sub
```

-STRUCT-  
 -Sequence-  
 ""  
 -INTERFACE-  
 -INTERFACE-  
 -INTERFACE-  
 -INTERFACE-  
 -INTERFACE-

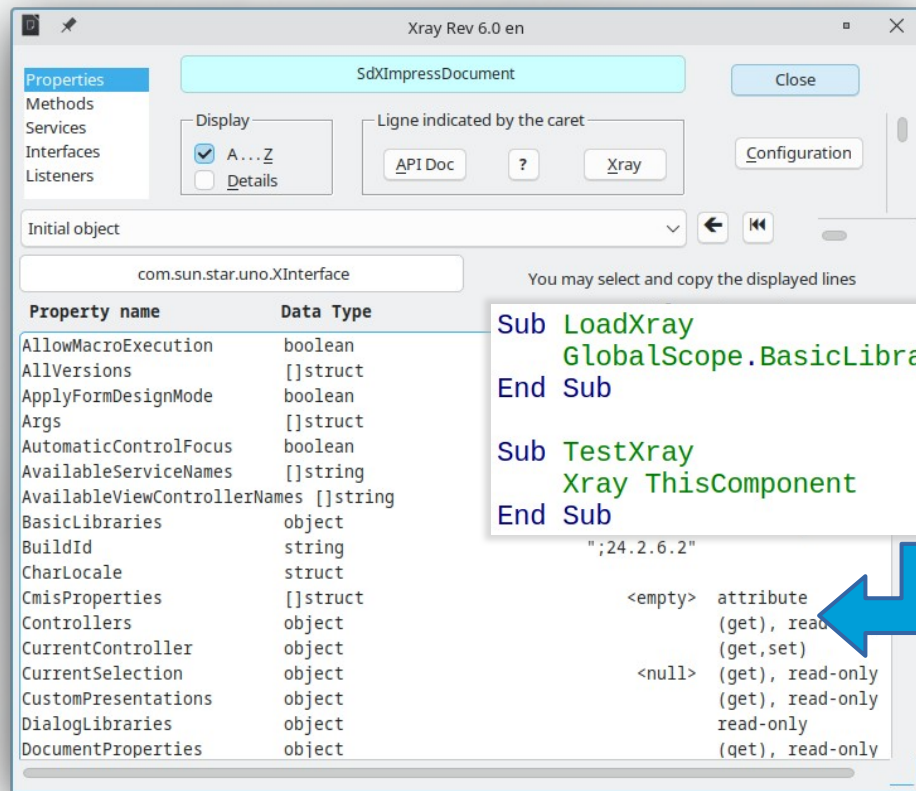


# Inspecting Objects

Use the MRI or Xray extensions to inspect objects and discover its properties and methods

Download Link:

[https://github.com/rafaelhlma/basic\\_calc\\_tutorial/blob/main/external/XrayTool60\\_en.odt](https://github.com/rafaelhlma/basic_calc_tutorial/blob/main/external/XrayTool60_en.odt)



# Spreadsheet Macros

Example of a Calc Macro that generates the Z distribution

	A	B	C	D
1	Z-Value	P(Z < z)		
2	-3	0,001349898		
3	-2,5	0,006209665		
4	-2	0,022750132		
5	-1,5	0,066807201		
6	-1	0,158655254		
7	-0,5	0,308537539		
8	0	0,5		
9	0,5	0,691462461		

Generate Z Table

```

Sub CreateStdNormalDistrTable
    Dim oSheet As Object, oRange As Object, oCell As Object
    oSheet = ThisComponent.Sheets(0)
    oRange = oSheet.getCellRangeByName("A1:B1")
    oRange.setDataArray(Array(Array("Z-Value", "P(Z < z)")))
    Dim zValue As Double, nRow As Integer, sAddress As String
    nRow = 1
    For zValue = -3 To 3 Step 0.5
        oCell = oSheet.getCellByPosition(0, nRow)
        sAddress = oCell.AbsoluteName
        oCell.setValue(zValue)
        oCell = oSheet.getCellByPosition(1, nRow)
        oCell.setFormula("=NORM.S.DIST(" & sAddress & ";1)")
        nRow = nRow + 1
    Next zValue
End Sub
    
```

# Recording Macros

## Steps to record a macro

- Go to **Tools – Options – Advanced** and choose “Enable macro recording”
- A new command “Record Macro” will be available in the **Tools – Macros** menu
- Start recording the macro, make changes to the spreadsheet and click “Stop recording”
- Save the code to a new Sub

# Recording Macros

Save the generated macro: recorded macros use UNO dispatch commands

```
sub FormatCell
rem -----
rem define variables
dim document as object
dim dispatcher as object
rem -----
rem get access to the document
document = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
rem -----
dim args1(0) as new com.sun.star.beans.PropertyValue
args1(0).Name = "Bold"
args1(0).Value = true

dispatcher.executeDispatch(document, ".uno:Bold", "", 0, args1())
```



# Recording Macros

## What can we do with the macro?

- Run it via Tool – Macros – Run Macro
- Associate the macro with a shape or control
- Add the macro to a menu
- Associate the macro with a shortcut
- Add a toolbar entry with the macro

# Dispatching UNO Commands

## LibreOffice offers various UNO commands

- Visit the link below for a list of all commands  
<https://wiki.documentfoundation.org/Development/DispatchCommands>
- Read more about DispatchHelper here  
[https://api.libreoffice.org/docs/idl/ref/interfacecom\\_1\\_1sun\\_1\\_1star\\_1\\_1frame\\_1\\_1XDispatchHelper.html](https://api.libreoffice.org/docs/idl/ref/interfacecom_1_1sun_1_1star_1_1frame_1_1XDispatchHelper.html)

# Dispatching UNO Commands

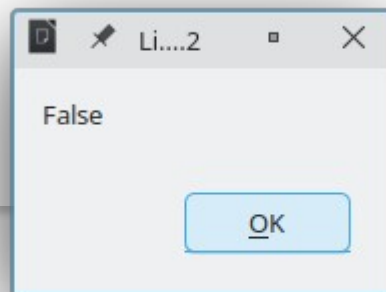
```
Sub CopyPaste
    Set oDispatch = CreateUnoService("com.sun.star.frame.DispatchHelper")
    Set oFrame = StarDesktop.ActiveFrame
    ' Select cells B3 to B6
    Dim args1(0) As New com.sun.star.beans.PropertyValue
    args1(0).Name = "ToPoint"
    args1(0).Value = "B3:B6"
    oDispatch.executeDispatch(oFrame, ".uno:GoToCell", "", 0, args1)
    ' Copy
    oDispatch.executeDispatch(oFrame, ".uno:Copy", "", 0, Array())
    ' Go to cell D3
    Dim args2(0) As New com.sun.star.beans.PropertyValue
    args2(0).Name = "ToPoint"
    args2(0).Value = "D3"
    oDispatch.executeDispatch(oFrame, ".uno:GoToCell", "", 0, args2)
    ' Paste and deselect
    oDispatch.executeDispatch(oFrame, ".uno:Paste", "", 0, Array())
    oDispatch.executeDispatch(oFrame, ".uno:Deselect", "", 0, Array())
End Sub
```

# Using UNO Services

## Example: using the *SpellChecker* service

```
Sub IsCorrect(aWord As String) As Boolean
    Dim aLocale As New com.sun.star.lang.Locale
    aLocale.Language = "en"
    aLocale.Country = "US"
    oSpellChecker = CreateUnoService("com.sun.star.linguistic2.SpellChecker")
    IsCorrect = oSpellChecker.isValid(aWord, aLocale, Array())
End Sub

Sub CheckSpelling
    MsgBox IsCorrect("housee")
End Sub
```





## Example: getting *SpellChecker* suggestions

```
Sub CheckAlternatives
  Dim aLocale As New com.sun.star.lang.Locale
  aLocale.Language = "en"
  aLocale.Country = "US"
  oSpellChecker = CreateUnoService("com.sun.star.linguistic2.SpellChecker")
  xAlternatives = oSpellChecker.spell("housee", aLocale, Array())
  If Not IsNull(vAlternatives) Then
    vAlt = xAlternatives.getAlternatives()
    sAltList = vAlt(0)
    For i = 1 to UBound(vAlt)
      sAltList = sAltList & Chr(10) & vAlt(i)
    Next i
    MsgBox sAltList
  End If
End Sub
```

# Using UNO Services

## SpellChecker service

[https://api.libreoffice.org/docs/idl/ref/servicecom\\_1\\_1sun\\_1\\_1star\\_1\\_1linguistic2\\_1\\_1SpellChecker.html](https://api.libreoffice.org/docs/idl/ref/servicecom_1_1sun_1_1star_1_1linguistic2_1_1SpellChecker.html)

## Locale struct

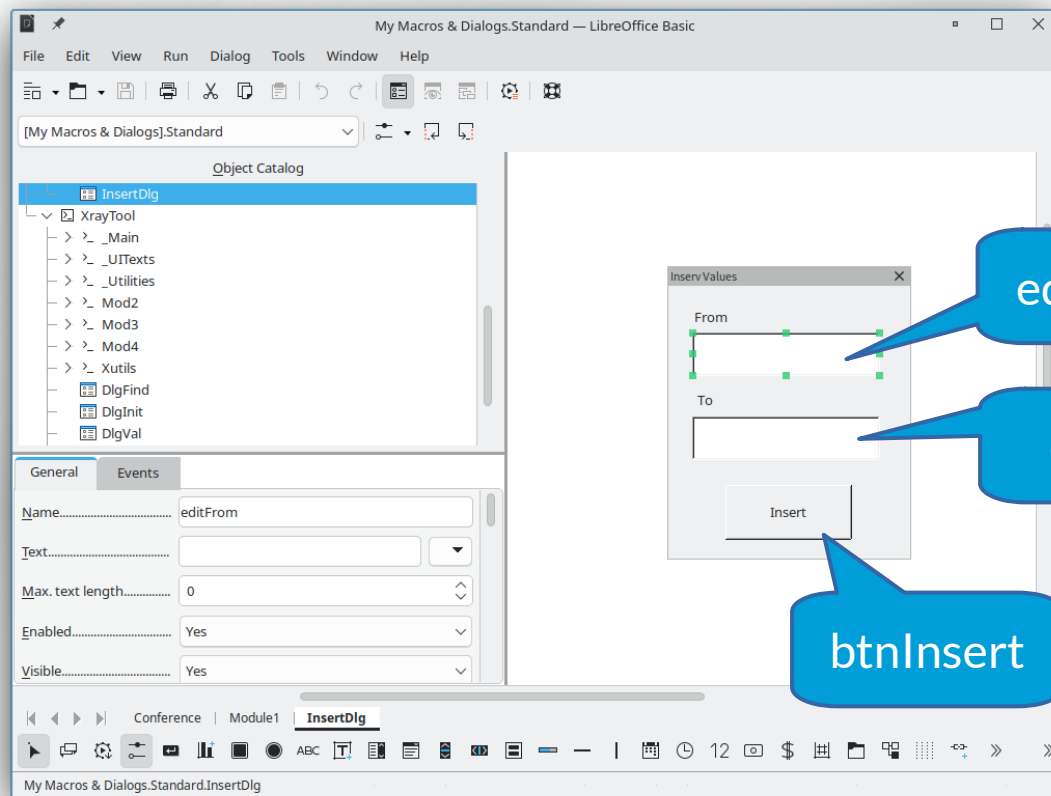
[https://api.libreoffice.org/docs/idl/ref/structcom\\_1\\_1sun\\_1\\_1star\\_1\\_1lang\\_1\\_1Locale.html](https://api.libreoffice.org/docs/idl/ref/structcom_1_1sun_1_1star_1_1lang_1_1Locale.html)

## XSpellAlternatives interface

[https://api.libreoffice.org/docs/idl/ref/interfacecom\\_1\\_1sun\\_1\\_1star\\_1\\_1linguistic2\\_1\\_1XSpellAlternatives.html](https://api.libreoffice.org/docs/idl/ref/interfacecom_1_1sun_1_1star_1_1linguistic2_1_1XSpellAlternatives.html)

# Creating Dialogs

Use the Dialog  
Editor to create  
a dialog



# Code to open the dialog and insert the numbers



```
Global oDlg As Object

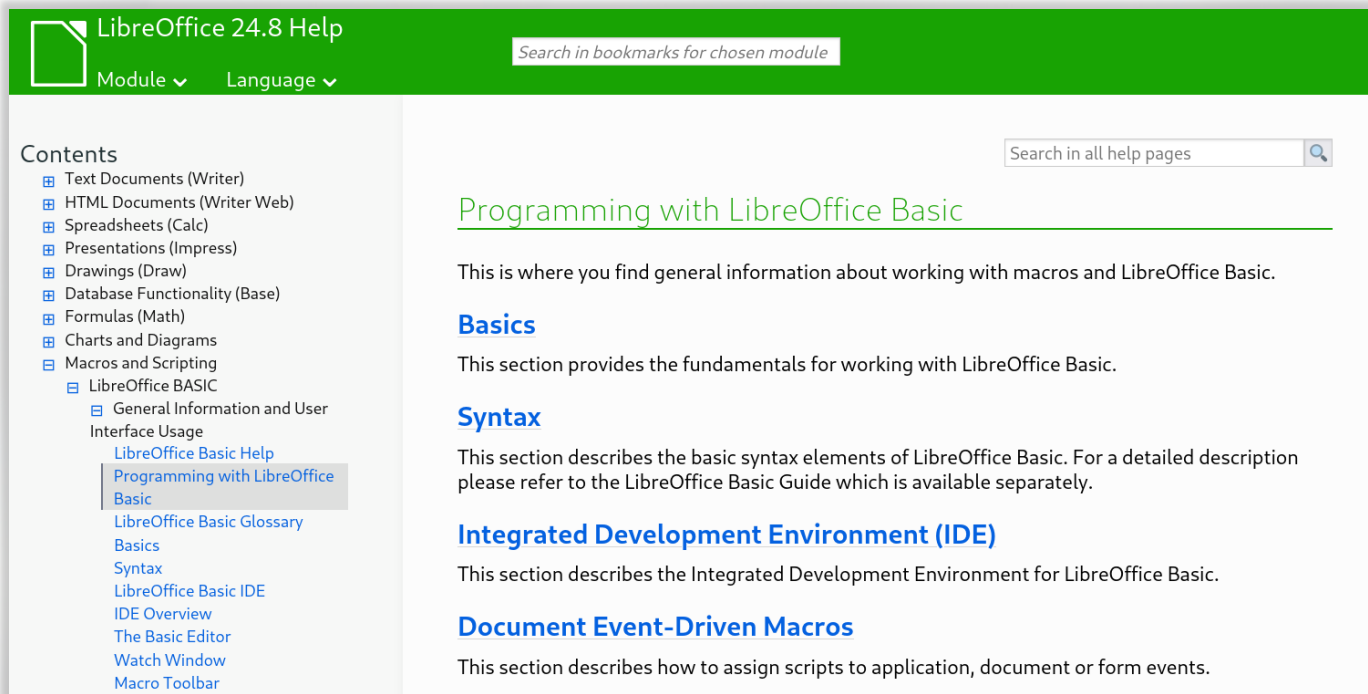
Sub ShowInsertDlg
    oDlg = CreateUnoDialog(DialogLibraries.Standard.InsertDlg)
    oDlg.execute()
End Sub

Sub BtnInsert_Execute(evt)
    ' Create array with values
    Dim arrValues() As Object
    nFrom = CInt(oDlg.getControl("editFrom").getText())
    nTo = CInt(oDlg.getControl("editTo").getText())
    nSize = nTo - nFrom + 1
    Redim arrValues(0 To nSize - 1) As Object
    For i = 0 To nSize - 1
        arrValues(i) = Array(nFrom + i)
    Next i
    ' Enter values in current cell
    oSel = ThisComponent.CurrentSelection
    oSheet = ThisComponent.CurrentController.ActiveSheet
    oRange = oSheet.getCellRangeByPosition(_
        oSel.CellAddress.Column, _
        oSel.CellAddress.Row, _
        oSel.CellAddress.Column, _
        oSel.CellAddress.Row + nSize - 1)
    oRange.setDataArray(arrValues)
End Sub
```



# More Learning Resources

## LibreOffice Basic official help pages




The screenshot shows the LibreOffice 24.8 Help interface. The top bar is green with the LibreOffice logo, version '24.8 Help', and a search box. Below the bar, there's a sidebar with a 'Contents' list. The main content area displays the title 'Programming with LibreOffice Basic' in green, followed by an introduction paragraph, and then sections for 'Basics', 'Syntax', 'Integrated Development Environment (IDE)', and 'Document Event-Driven Macros', each with a brief description.

**LibreOffice 24.8 Help**

Module ▾ Language ▾

**Contents**

- Text Documents (Writer)
- HTML Documents (Writer Web)
- Spreadsheets (Calc)
- Presentations (Impress)
- Drawings (Draw)
- Database Functionality (Base)
- Formulas (Math)
- Charts and Diagrams
- Macros and Scripting
  - LibreOffice BASIC**
    - General Information and User Interface Usage
      - LibreOffice Basic Help
        - Programming with LibreOffice Basic**
        - LibreOffice Basic Glossary
        - Basics
        - Syntax
        - LibreOffice Basic IDE
        - IDE Overview
        - The Basic Editor
        - Watch Window
        - Macro Toolbar



### Programming with LibreOffice Basic

This is where you find general information about working with macros and LibreOffice Basic.

#### Basics

This section provides the fundamentals for working with LibreOffice Basic.

#### Syntax

This section describes the basic syntax elements of LibreOffice Basic. For a detailed description please refer to the LibreOffice Basic Guide which is available separately.

#### Integrated Development Environment (IDE)

This section describes the Integrated Development Environment for LibreOffice Basic.

#### Document Event-Driven Macros

This section describes how to assign scripts to application, document or form events.

Link: <https://help.libreoffice.org/latest/en-US/text/sbasic/shared/01000000.html>

# More Learning Resources

## LibreOffice Developer's Guide

### LibreOffice Developer's Guide

Page [Discussion](#) [Read](#) [View source](#) [View history](#)

[< Documentation | DevGuide](#)

[Contributing to the Developer's Guide](#)

#### What This Manual Covers

This manual describes how to write programs using the component technology UNO (Universal Network Objects) with LibreOffice.

Most examples provided are written in Java. As well as Java, the language binding for C++, the UNO access for LibreOffice Basic and the OLE Automation bridge that uses LibreOffice through Microsoft's component technology COM/DCOM is described.

#### How This Book is Organized

Every page of this book has a Table of Contents at the right side of the page. This TOC shows the content of the current part and navigation links to browse parts and pages of this book.

#### First Steps

The First Steps chapter describes the setting up of a Java UNO development environment to achieve the solutions you need. At the end of this chapter, you will be equipped with the essentials required for the following chapters about the LibreOffice applications.

#### Professional UNO

This chapter introduces API and UNO concepts and explains the specifics of the programming languages and technologies that can be used with UNO. It will help you to write industrial strength UNO programs, use one of the languages besides Java or improve your understanding of the API reference.

#### LibreOffice Developer's Guide

- [First Steps](#)
- [Professional UNO](#)
- [Writing UNO Components](#)
- [Extensions](#)
- [Advanced UNO](#)
- [Office Development](#)
- [Text Documents](#)
- [Spreadsheet Documents](#)
- [Drawing Documents and Presentation Documents](#)
- [Charts](#)
- [LibreOffice Basic](#)
- [Database Access](#)
- [Forms](#)
- [Universal Content Broker](#)
- [Configuration Management](#)
- [JavaBean for Office Components](#)
- [Accessibility](#)
- [Scripting Framework](#)
- [Graphical User Interfaces](#)
- [Guidelines and](#)

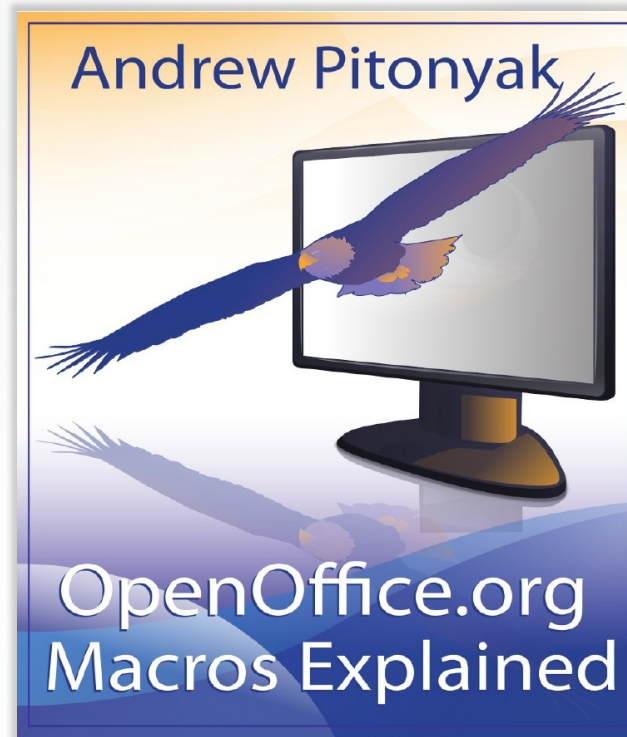
Link: [https://wiki.documentfoundation.org/Documentation/DevGuide/LibreOffice\\_Developers\\_Guide](https://wiki.documentfoundation.org/Documentation/DevGuide/LibreOffice_Developers_Guide)

# More Learning Resources

**Andrew Pitonyak's book**  
**"OpenOffice Macros Explained"**

Free to download at:

<https://www.pitonyak.org/oo.php>



*Thank  
you*

Questions?