



OOoCon 2005



Introduction to OpenOffice.org macro development



Agenda

- Who am I
- Who are you
- Where does this BoF fit
- OOo extensions wiki
- Recorded macros / dispatch calls
- Getting started
- Q & A



Who am I

- Ian Laurensen
- Wrote initial version of the Migration Guide
 - http://oooauthors.org/en/authors/userguide2/migration/draft_pub/
- lannz macros
 - <http://homepages.paradise.net.nz/hillview/OOo/>
- Maintainer of wiki on OOo macro development
 - <http://ext.openoffice.org.nz/>



Have you:

- Recorded macros in OOo?
- Written macros in OOo?
- Written macros in VBA?
- Programmed in any language?



Where does this BoF fit?

- Aim to help people to extend OOo functionality
- Macros could be packaged into UNO packages as per Paolo Mantovani's talk
- Is part of the skill set required for Addons: From production to end-users as per Laurent Godard tomorrow



What is a macro?

- A small program that extends the functionality of OpenOffice.org
- Many people associate "macros" with Microsoft Office viruses thus I'll use the term "Extension"
- Technically an extension can be more than just a macro
- An OpenOffice.org extension may be used to:
 - Automate a repetitive task
 - Integrate other applications with OpenOffice.org
 - Implement feature found in another application



Wiki

- Wiki for OOo extensions contains:
 - A getting started guide
 - A cookbook of generic techniques
 - A cookbook of some Writer & Calc techniques
 - A place for extension collaboration
 - The on-line Basic help files
 - The API documentation
- <http://ext.openoffice.org.nz>
- Needs to be extended and improved
 - Nothing for Impress, Draw or Base yet



Why?

- Why a wiki?
 - Easily editable by all
 - Information rather than debate
 - Code explanations as long as required
 - Easy cross referencing
- Why repeat info from elsewhere?
 - Original info could be improved
 - Ease of cross referencing within wiki



Languages

- OpenOffice.org has a Macro language (OOo Basic)
- Other language bindings include:
 - Python (included in OOo but no IDE)
 - Java
 - C++
 - Common Language Infrastructure (CLI)
 - Object Linking and Embedding (OLE) Automation
- This presentation only looks at OOo Basic



Why OOo Basic?

- All users of OpenOffice.org have access to it
 - IDE
 - OpenOffice.org provides the Basic interpreter
- Full access to all of the API
- Easier to test out an idea before implementing in another language
- On-line documentation
- Based on Microsoft's VBA
 - Which a lot of people already know
- My experience



Macro recorder and UNO dispatch

- Recorded code
- Tidying recorded code
- Better approach





Macro recorder

- The following slides all do the same task in Writer
 - *Type “Some text.”*
 - *Insert a paragraph mark*
 - *Type “A new paragraph with a ”*
 - *Turn bold on*
 - *Type “bold”*
 - *Turn bold off*
 - *Type “ word in it”.*

Some text.

A new paragraph with a **bold** word in it.

Recorded version

```
sub Example
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 = "Text"
args1(0).Value = "Some text."

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

rem -----
dispatcher.executeDispatch(document, ".uno:InsertPara", "", 0, args1())

rem -----
dim args3(0) as new com.sun.star.beans.PropertyValue
args3(0).Name = "Text"
args3(0).Value = "A new paragraph with a "

dispatcher.executeDispatch(document, ".uno:InsertText", "", 0, args3())

rem -----
dim args4(0) as new com.sun.star.beans.PropertyValue
args4(0).Name = "Bold"
args4(0).Value = true

dispatcher.executeDispatch(document, ".uno:Bold", "", 0, args4())

rem -----
dim args5(0) as new com.sun.star.beans.PropertyValue
args5(0).Name = "Text"
args5(0).Value = "bold"

dispatcher.executeDispatch(document, ".uno:InsertText", "", 0, args5())
```

```
rem -----
dim args6(0) as new com.sun.star.beans.PropertyValue
args6(0).Name = "Bold"
args6(0).Value = false

dispatcher.executeDispatch(document, ".uno:Bold", "", 0, args6())

rem -----
dim args7(0) as new com.sun.star.beans.PropertyValue
args7(0).Name = "Text"
args7(0).Value = " word in it."

dispatcher.executeDispatch(document, ".uno:InsertText", "", 0, args7())

end sub
```



```
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 = "Text"
args1(0).Value = "Some text."

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



Tidied recorded code

Sub example

```
fnDispatch("InsertText", array("Text","Some text."))
fnDispatch("InsertPara")
fnDispatch("InsertText", array("Text","A new paragraph with a "))
fnDispatch("Bold", array("Bold",true))
fnDispatch("InsertText", array("Text","bold"))
fnDispatch("Bold", array("Bold",false))
fnDispatch("InsertText", array("Text"," word in it."))
end sub
```

- For a list of "dispatch commands see:
 - <http://api.openoffice.org/files/documents/22/1344/slots.sxc>
 - <http://www.openoffice.org/files/documents/25/2570/commandsReference.html>



My fnDispatch



```
function fnDispatch(sCommand as string, optional mArgs)
oFrame = ThisComponent.getCurrentController.getFrame
oDispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
'on error resume next
if isMissing(mArgs) then
    fnDispatch = oDispatcher.executeDispatch(oFrame, ".uno:" & sCommand, "", 0, array())
else
    nArgs = uBound(mArgs) \ 2
    dim Args(nArgs) as new com.sun.star.beans.PropertyValue
    for i = 0 to nArgs
        Args(i).name = mArgs(i * 2)
        Args(i).value = mArgs(i * 2 + 1)
    next
    fnDispatch = oDispatcher.executeDispatch(oFrame, ".uno:" & sCommand, "", 0, Args())
end if
end function
```


Using API calls

```
sub Example
oVC = thisComponent.getCurrentController.getViewCursor
oText = ...
oText.insertString(oVC, "Some text", False)
oText.insertControlCharacter(oVC, _
    com.sun.star.text.ControlCharacter.PARAGRAPH_BREAK, False)
oText.insertString(oVC, "First paragraph with ...", False)
oVC.setPropertyValue("CharWeight", com.sun.star.awt.FontWeight.BOLD)
oText.insertString(oVC, "bold", False)
oVC.setPropertyValue("CharWeight", com.sun.star.awt.FontWeight.NORMAL)
oText.insertString(oVC, " word in it.", False)
end sub
```



Getting started

- Where is code stored
 - The IDE - Integrated Development Environment
 - Saving code
 - Naming Conventions
 - Xray tool
 - SDK
 - Terminology
 - Common starting points
 - Introspection
 - Working with objects
 - Creating objects
- 
- 



Where to write code?

- Basic code is stored in modules within libraries
- A library can be
 - *An OpenOffice.org library*
 - *Shared (for a network install)*
 - *Just for the current user*
 - *Within a document or template*
 - *Code only available when that document is open.*
- OpenOffice.org libraries
 - *Not in a document or template*
 - *To find actual location*
 - *Tools > Options... > OpenOffice.org > Paths > BASIC*



Accessing the IDE

- To access the IDE for the first time:
 - *OpenOffice.org 1.1.x: Tools > Macros > Macro... >*
 - *OpenOffice.org 1.9.x:
Tools > Macros > Organize macros > OpenOffice.org
Basic... >*
- To get started we will use Module1 in the Standard library which is for current user only
 - *Type a name for the new macro: HelloWorld*
 - *In the macro from listbox select: Standard*
 - *Click New*
- Now that IDE is open use either
 - *Window menu*
 - *Task bar*

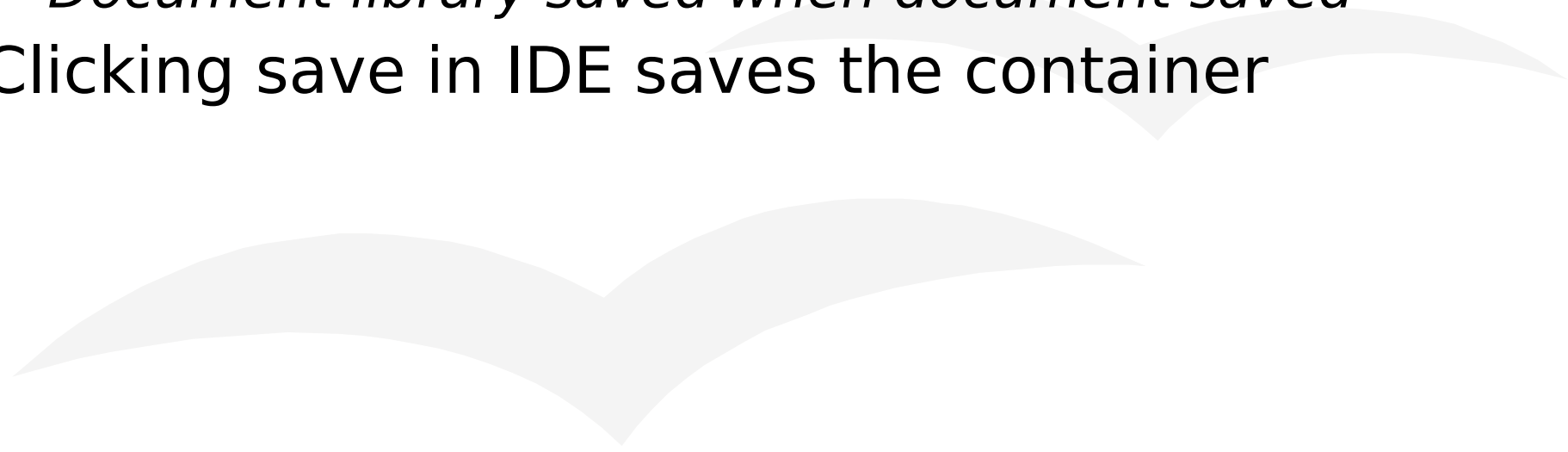


Demonstrate

- Windows in IDE
- Controls in IDE
- Editing in IDE
- Running code
 - *Directly from the IDE*
 - *From the tools menu*
 - *(Version 1.1.x) Tools > Macros > Macro...*
 - *(Version 1.9.x) Tools > Macro > Run Macro...*
 - *Assigning the macro to a key press*



Saving

- Code saved when container saved
 - *OpenOffice.org library saved on exit*
 - *Document library saved when document saved*
 - Clicking save in IDE saves the container
- 



Naming conventions

- Variables - First Letter
 - *Lowercase*
 - *Denotes the type of value variable will hold*
 - *b Boolean (TRUE or FALSE)*
 - *f (Float) single / double / Currency*
 - *m (matrix) Array*
 - *n Integer / long*
 - *o Object, service, or interface*
 - *s String*
 - *v Variant, Any*



Naming conventions (Continued)

- Use long descriptive names
- Use CamelCase
- Exceptions
 - *Simple counters - typically use i, j, k*
 - *Building a string - typically use s*

Naming conventions (Continued again)

- Subs and Functions
 - Starting point has descriptive name
 - E.g. Sub SetupAltKeyHandler
 - Any sub that gets called by the code the name starts with "sub"
 - E.g. Sub subSaveAltKeySettings
 - Subs that get called by an event start with "subEvent"
 - E.g. Sub subEventOkInitiated
 - Functions start with "fn"
 - E.g. Function fnWhichComponent



X-Ray tool

- X-Ray tool by Bernard Marcelly is a must!
 - See oocomacros.org/dev.php#101416
- A run-time equivalent of code completion
- Demonstration of installing a macro library using X-Ray tool as an example
- API documentation accessed from the X-ray tool
- API documentation in SDK so need to download SDK



SDK

- Download and extract the sdk
 - Should be
http://www.openoffice.org/dev_docs/source/sdk/index.html
 - But more recent often found in mirrors
 - Choose one of the FTP mirrors for developers listed in:
<http://download.openoffice.org/680/index-nojs.html>
 - Remove stuff in the URL to the right of: developer/
 - Look for latest 680_mxx_sdk/
- There are instructions for installing SDK
 - For OOo Basic this is not necessary



API terms

- **Interface** is a definition of a set of methods (and their arguments) that a service which implements that interface must have
- A **service** contains methods
- **Methods** are routines that do something
- **Properties** are values
- **Modules** group related interfaces and services
- All services start with the name “com.sun.star” then the name of the module then the name of the service



Common starting points

- StarDesktop
 - ThisComponent
 - getCurrentSelection
- 
- 



StarDesktop

- New documents

```
function fnNewDoc(sDocType as string)
fnNewDoc = StarDesktop.loadComponentFromURL(_
  "private:factory/" & sDocType , "_blank", 0, Array())
end function
```

```
oDoc = fnNewDoc("swriter")
```

- Open documents

```
function fnOpenDoc(sFile)
sURL = ConvertToURL(sFile)
fnOpenDoc = StarDesktop.loadComponentFromURL(sURL, _
  "_blank", 0, Array())
end function
```

```
' Linux
```

```
oDoc = fnOpenDoc("/home/danny/Desktop/MyCalc.sxc")
```

```
' Windows
```

```
oDoc = fnOpenDoc("c:\MyCalc.sxc")
```



ThisComponent getCurrentSelection

- ThisComponent = the currently open document
- getCurrentSelection
 - Common to work with the current selection
 - Objects that "ThisComponent" returns have method "getCurrentSelection"

```
oCurSelection = thisComponent.getCurrentSelection
```

- Check selected object has service that contains the method that we want to apply to the object

```
if oCurSelection.supportsService( _  
    "com.sun.star.text.TextRanges" ) then
```



Introspection

- **HasUnoInterfaces**


- A function provided by OOo Basic
- Returns true if all of the specified interfaces are available for the specified object

```
if HasUnoInterfaces(oDoc, _  
    "com.sun.star.lang.XServiceInfo") then
```

- Most OpenOffice.org objects provide the method `supportsService` because they have the interface `com.sun.star.lang.XServiceInfo`
- **supportsService** is a method which returns true if the specified service is available



Working with objects

- Properties
 - Enumerations
 - Named and / or index access
- 



Properties

```
oCurSelection = thisComponent.getCurrentSelection()  
if oCurSelection.supportsService  
  _("com.sun.star.text.TextRanges") then  
  nCount = oCurSelection.Count  
  'xray.xray oCurSelection.getByIndex(0)  
  for i = 0 to nCount - 1  
    oCurSelection.getByIndex(i).setPropertyValue( _  
      "CharStyleName", "Strong Emphasis")  
  next  
end if
```



Enumeration

```
' Create enumeration object
oTextElementEnum = thisComponent.getText().createEnumeration
'or
  thisComponent.getCurrentSelection().getByIndex(i).createEnumeration
' loop over all text elements
while oTextElementEnum.hasMoreElements
  oTextElement = oTextElementEnum.nextElement
  if oTextElement.supportsService("com.sun.star.text.TextTable") then
    MsgBox "The current block contains a table."
  else ' oTextElement.supportsService("com.sun.star.text.Paragraph")
    MsgBox "The current block contains a paragraph."
  end if
wend
```



Named and / or index access

- Index access

```
oSheet = thisComponent.getSheets.getByIndex(0)
```

- Named access:

```
if thisComponent.getSheets.hasByName("Sheet1") then  
  oSheet =  
    thisComponent.getSheets.getByName("Sheet1")
```

- To loop through all the available object names


```
mNames = thisComponent.getSheets.getElementNames  
for i = lbound(mNames) to ubound(mNames)  
  msgbox mNames(i)  
next
```

- insertByName, replaceByname,
 removeByName

```
thisComponent.getSheets.insertByName("NewSheet")
```



Creating new objects

- Create method
 - CreateInstance
 - createUnoService
- 



Create Method

- Some objects provide methods for creating another object

- *E.g. Creating a text cursor*

```
oVC = thisComponent.getCurrentController.getViewCursor
```

```
oCursor = oVC.getText.createTextCursorByRange(oVC)
```

```
'Demonstrate that new cursor not visible
```

```
oCursor.gotoStartOfSentence(false)
```

```
oCursor.gotoEndOfSentence(true)
```

```
msgbox oCursor.getString
```

createInstance

- Some objects are context dependent but don't have their own create method

! To add a rectangle to the first page of a drawing document

```
dim aPoint as new com.sun.star.awt.Point
```

```
dim aSize as new com.sun.star.awt.Size
```

```
aPoint.X = 1000
```

```
aPoint.Y = 1000
```

```
aSize.Width = 10000
```

```
aSize.Height = 10000
```

```
oRectangleShape = thisComponent.createInstance(  
    "com.sun.star.drawing.RectangleShape")
```

```
oRectangleShape.Size = aSize
```

```
oRectangleShape.Position = aPoint
```

```
thisComponent.getDrawPages.getByIndex(0).add(oRectangleShape)
```



createUnoService

- Some objects are context independent
 - To create these objects use the OpenOffice.org Basic command createUnoService
- For example to create the equivalent to StarDesktop

```
oDesktop = createUnoService("com.sun.star.frame.Desktop")
```


Uno Structs

- Remember this line

```
dim aPoint as new com.sun.star.awt.Point
```

- dim is one way of creating a variable that contains a UNO struct

- Another is

```
aPoint = createUnoStruct("com.sun.star.awt.Point")
```

- When declaring UNO structs case is important
 - Everything up to the name of the struct is lowercase
 - Name of the struct is in TitleCase.

UNO constants

- Constant groups (enumerations) have fixed set of values
- Each value can be represented by UNO constant

• For example:

`com.sun.star.awt.KeyModifier.MOD2`

- OpenOffice.org constants are case sensitive
 - Everything up to and including the module name is always lowercase
 - The constant group is in TitleCase
 - The actual constant name is all UPPERCASE

How do I know how to access or create an object?

- Does the object already exist if so access it from something like thisComponent
- Will the new object belong to another object, if so does the owner have a specific method for creating the object, if so use it
- The new object will belong to another object, but that object doesn't provide a specific method for creating it, but does provide createInstance
- If the object doesn't provide createInstance are you sure you are using the correct object, or is it context independent



Questions Discussion Problem solving



Thanks

- Thank you!
- For getting me here
 - *Sun*
- For hosting and assistance with setting up wiki
 - *Dave Lane & Jim Cheetham of egressive.com*
- For the ext.openoffice.org.nz domain name
 - *Chris Maffey of maffey.com*
- For offering to host wiki
 - *Justin Clift of digitaldistribution.com*