OpenOffice Basic

Simple OpenOffice Macros

One of the major complaints heard from Word users moving to OpenOffice is the fact that they can't create even the simple kind of macro. In this article we will be looking at organizing, importing, and applying simple macros with OpenOffice Basic. BY IAN TRAVIS

Adding a Basic style programming language to an office suite has its pros and cons. On the one hand macros can make life so much easier for the user, on the other, some of the most devastating attacks on desktop operating systems in the past have used Basic programming languages. Let's leave that side of the equation to the Gray Hats and take a look at the user benefits instead.

Getting Organized

If you are new to OpenOffice macros, the *Tools/Macro* menu is probably the best place to start, as this is where OpenOffice organizes its macro facilities. Those of you expecting to find a macro recorder are in for a disappointment, there isn't one.

Selecting *Tools/Macro* opens the fairly cryptic looking *Macro* dialog box. The box is divided into three areas. The panel on the right under *Macro from* contains a list of libraries and modules, more specifically the *soffice* collection containing the *Euro*, *FormWizard*, *Gimmicks*, *ImportWizard*, *Schedule*, *Standard*, *Template*, *Tools*, and *WebWizard* libraries. Any documents you opened before accessing the *Macro* dialog box are shown at the bottom of the list, and will typically contain an entry for the standard module.

Double-clicking a library such as *Euro* expands the list of modules in that library allowing you to select a module and display a list of the macros it contains. The *Macro name* box shows the name of the macro currently highlighted in the list. Figure 1 shows the *Macro* dialog box including an *soffice* macro called *InsertStringToCell*, as well as an OpenOffice text document called *Life with OpenOffice macros*.

After selecting a macro, you can use the buttons on the right of the dialog box to *Run*, *Close*, *Assign*, *Edit* or *Delete* the macro. Clicking on *Assign* opens the *Configuration* dialog box which allows you to add your own macros to menus, define keyboard shortcuts, add macros to toolbars, or assign an event to a macro.

OpenOffice Macro Organizer

Clicking on the *Organizer* button launches the OpenOffice Macro Organizer and displays the *Modules* tab, where you can create new modules or dialogs, and edit existing modules (indicated by a sheet icon) or dialogs (indicated by a green rectangle). Clicking on the *Libraries* tab allows you to select either the *soffice* category or an active document in the *Application/Document* list box and then add new libraries, append to an existing library, edit or assign passwords for existing libraries, or even delete a library (with the exception of *Standard*).

Figure 3 shows how a new library for a current document is created using the *New Library* dialog box.

Your Very Own Macro

Let's take a look at a simple macro available from the download section of the *OO Extras* website at [1]. The *Transpose* macro switches two letters within a word, assuming that the cursor is placed between those characters. Listing 1 contains the source code.

Let's assume you want to add this macro to the *Standard* library in order to apply it to documents you edit with OpenOffice. After downloading the macro file, select *Tools/Macro*, click on the *Standard* library below *Macro from*, type a name for the macro in the text box below *Macro name*, then click on *New* and type a name for the module you will be creating; in our case this will be *Transposer* (see Figure 2). A macro called *Main* is automatically created if you forget to supply the macro name, you can delete this empty macro after creating a macro of your own.

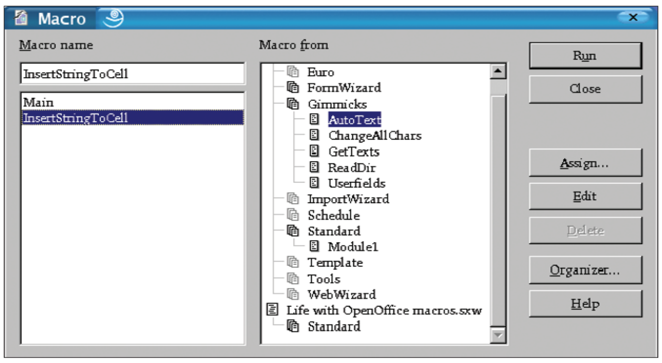


Figure 1: The Macro dialog box in OpenOffice

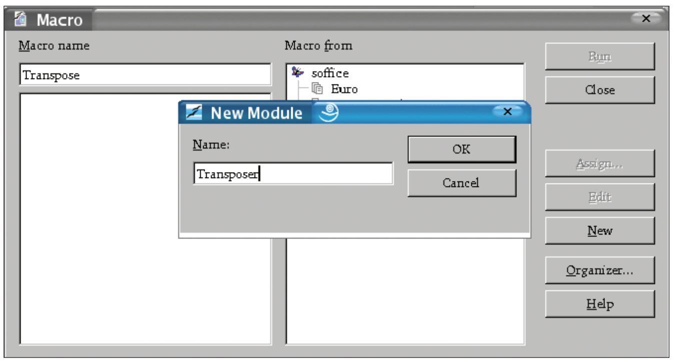
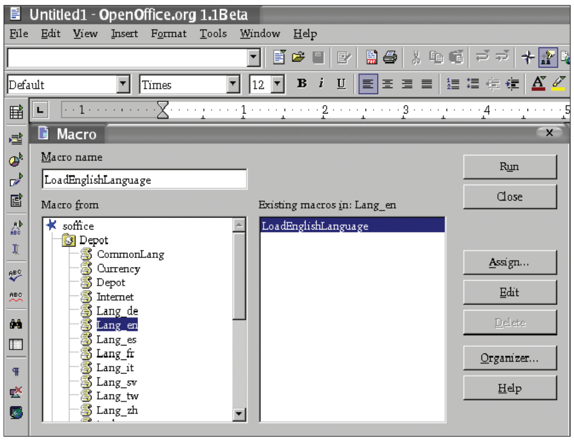


Figure 2: Specifying names for the new macro and the new module


Untitled1 - OpenOffice.org 1.1Beta
File Edit View Insert Format Tools Window Help
Default
Times
12
Macro
Run
Close
Assign...
Edit
Delete
Organizer...
Help

Macro name: LoadEnglishLanguage

Macro from: soffice, Depot, CommonLang, Currency, Depot, Internet, Lang_de, Lang_en, Lang_es, Lang_fr, Lang_it, Lang_sv, Lang_tw, Lang_zh

Existing macros in: Lang_en: LoadEnglishLanguage

Figure 4: Improved Macro Management in OpenOffice 1.1

Editing a Macro

After creating the new macro, you can click on *Edit* to launch the OpenOffice macro development environment, which should display an empty framework like the following for the macro:

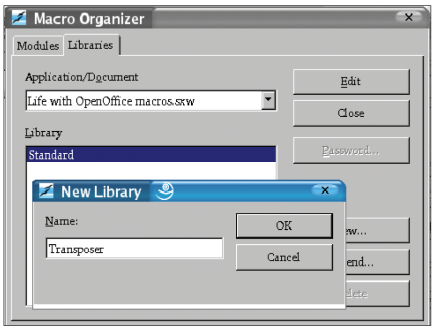
```
REM  *****  BASIC  *****

Sub Transpose

End Sub
```

The source code for the macro needs to be inserted between the *Sub* and *End Sub* tags. OpenOffice provides a function for this purpose; thus you can click on the second to last button in the macro bar, *Insert Source Text*, navigate to a Basic file of your choice in the dialog box that appears (this is transpose.bas in our example), select the file, and click on *Open* to insert the source code at the current cursor position. Of course a simple cut & paste from your favorite editor would do the job just as well.

As this is a tried and trusted macro, there is no need to edit the source code, so let's just close the developer environment and return to the text to try out the macro.



Application/Document: Life with OpenOffice macros.sxw

Library: Standard

Name: Transposer

Figure 3: Creating a new library for the current document

Running Macros the Hard Way

We should be able to run the macro in a document after completing these steps, so let's look for an appropriate typo that needs transposing. Purely by coincidence, the word "macro" has been misspelled as "marco" in a subtitle of the current document.

After placing the cursor between the "r" and the "c" we can click on *Tools/Macro* to open the *Macro* dialog box, select the *Transposer* module, and run the *Transpose* macro.

So far, so good

Of course the macro did what we expected it to do, but running it took about ten times as long as manually correcting the typo. This is where the *Assign* function we mentioned earlier comes into play. It is a good idea to assign a keyboard shortcut to this particular macro using *CTRL-T* or a similarly intuitive sequence without a current assignment in your OpenOffice version.

Assigning a Keyboard Shortcut

Open the *Macro* dialog box, select the *Transpose* macro, and then click on *Assign* to open the *Configuration* dialog box. Open the *Keyboard* tab, and locate the shortcut you want to assign in the list below *Shortcut keys*. Ensure that the shortcut does not have a previous assignment. Locate the module containing the

Listing 1: Transpose Macro

```
'
' Released by www.ooextras.org under the LGPL license.
' See http://www.ooextras.org/license.txt for terms of license
' Original homepage:
http://www.darwinwars.com/lunatic/bugs/oo_macros.html
'
Sub transpose
Dim oDocument, oDesktop as Object
Dim oText as Object
Dim oVCursor, oCursor As Object
Dim sWombat as string
' the two following lines get the active document
oDesktop = createUnoService("com.sun.star.frame.Desktop")
oDocument= oDesktop.getCurrentComponent()
oText = oDocument.Text
' after this, an obscure call gets the current cursor position
' but most cursor methods don't work with a view cursor
oVCursor = oDocument.currentcontroller.getViewCursor()
' so now I create an invisible cursor under it,
oCursor = oText.createTextCursorByRange(oVCursor.getstart())
' grab the next character
oCursor.goRight(1,TRUE)
' save a copy (no clipboard)
sWombat = ocursor.getString()
' delete the original (seems to be no obvious delete method)
' obviously, what I wanted was a method to cut to the clipboard
' but this doesn't seem possible
oCursor.setString("")
' next two lines move the invisible cursor
oCursor.CollapseToStart()
oCursor.goLeft(1,true)
' and now insert the cut character
oText.insertString(oCursor.getStart(),sWombat,false)
oVCursor.GoLeft(1,false) ' finally return the visible cursor whence it
came
End Sub
```

macro in the *Category* list below *Functions*, and select the library (*Transposer* in our case) to display a list of macros below *Function list*. Finally, click on *Modify* to apply the keyboard assignment. The shortcut key in the list will now have an entry for the macro. Click on *OK* to close the dialog box, and then on *Close* to return to the editing window. Pressing CTRL-T in the document will now run the *Transpose* macro.

Customizing Toolbars

Assigning keyboard shortcuts to macros is fine in some situations, but in others you might prefer to add the macros from a specific module to the menu. To do so, open the *Macro* dialog box as described earlier, click on a module and then on *Assign*, and select the *Menu* tab. Select the module containing the macro that you want to add to the menu in the list under *Functions*, select the menu item below which the macro should be added, and then click on the *New* button to add the macro.

Alternatively, you might like to add a collection of macros to a toolbar. To do so, open the *Macro* dialog box as described earlier, click on a module and then on *Assign*, and select the *Toolbars* tab. We will be customizing the function bar in this example.

To do select, select *Function bar* as the toolbar type and click on *Customize*, navigate to the macro you want to add to the function bar and select it to display an icon for the macro under *Icons*. Now drag the icon to the function bar and drop it.

Things to Come

Searching the Internet for Howtos and other guides on OpenOffice Basic programming will tend to confirm the rumors you may already have heard: the whole topic is under-documented. Not to worry: OpenOffice 1.1, which is currently at the beta stage, provides a solution to this dilemma. It tidies up the *Macro* dialog box, placing the libraries in the left-hand panel, and the individual macros on the right, to provide a more intuitive tree view (see Figure 4), and toolbar manipulation is vastly simplified by the addition of a split panel view.

Most importantly, OpenOffice 1.1 introduces a macro recorder, opening up the world of macro programming to Open Source converts. Listing 2 shows the OpenOffice 1.1 macro code for a Transpose function equivalent to the macro discussed earlier.

Conclusion

Although OpenOffice 1.0.x fully supports macro functionality, macro programming for it is far from being a pleasurable experience due to the lack of documentation. Help is on its way, and for the brave at heart just a 60 MB download away at the download site [2].

The addition of macro recording facilities to OpenOffice can justifiably be regarded as a major step forward for this major contender for the Office suite crown. ■

INFO

[1] OO Extras Website: *http://ooextras.sourceforge.net*

[2] OpenOffice beta download: *http://www.openoffice.org/dev_docs/source/1.1beta/index.html#linux*

Listing 2: Transpose macro in OpenOffice 1.1 code version

```
sub Transpose
rem define variables
dim document   as object
dim dispatcher as object
rem get access to the document
document   = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
dim args1(1) as new com.sun.star.beans.PropertyValue
args1(0).Name = "Count"
args1(0).Value = 1
args1(1).Name = "Select"
args1(1).Value = true

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

dispatcher.executeDispatch(document, ".uno:Cut", "", 0, Array())

dim args3(1) as new com.sun.star.beans.PropertyValue
args3(0).Name = "Count"
args3(0).Value = 1
args3(1).Name = "Select"
args3(1).Value = true

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

dim args4(1) as new com.sun.star.beans.PropertyValue
args4(0).Name = "Count"
args4(0).Value = 1
args4(1).Name = "Select"
args4(1).Value = false

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

dim args5(1) as new com.sun.star.beans.PropertyValue
args5(0).Name = "Count"
args5(0).Value = 1
args5(1).Name = "Select"
args5(1).Value = false

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

dispatcher.executeDispatch(document, ".uno:Paste", "", 0, Array())

end sub
```