

Comment Utiliser des Macros Basic dans OpenOffice.org

Diffusé par

Le Projet de Documentation OpenOffice.org

Table des Matières

Introduction — Qu'est-ce qu'une Macro?.....	3
Comment Trouver des Informations sur les Macros.....	5
Utilisation de la Boîte de Dialogue Macro.....	6
Utilisation du Gestionnaire de Macros.....	8
Création d'un Nouveau Module.....	8
Création d'une Nouvelle Bibliothèque.....	10
Ecrire une Macro.....	12
Exécution d'une Macro au sein de l'EDI.....	15
Sélectionner des Bibliothèques dans l'EDI.....	16
Appel d'une Macro Depuis une Autre Macro.....	18
Appeler une Macro avec un Bouton.....	18
Exemple avec Deux Boutons et une Etiquette.....	23
Appeler une macro Depuis un Événement.....	28
Appeler une Macro depuis le Clavier.....	29
Appeler une Macro avec un Bouton de Barre d'Outils.....	31
Appeler une Macro avec un Lien Texte.....	34
Appeler une Fonction Macro depuis une Feuille de Calcul.....	35
Appeler une Macro dans une Autre Bibliothèque.....	37
Appeler une Macro dans une Validation de Données.....	39
Création d'une Boîte de Dialogue.....	43
Appeler l'Interface de Programmation d'Applications depuis d'autres Langages.....	46
D'autres Exemples.....	47
Sommaire.....	54
Credits.....	55
Licence.....	55

Introduction – Qu'est-ce qu'une Macro?

Ce document se veut une démonstration du travail avec des macrocommandes OpenOffice.org BASIC, comment les organiser en modules et bibliothèques, et comment les appeler de différentes manières.

Ce document n'a pas pour but de vous introduire à la programmation du langage BASIC d'OpenOffice.org (préférez pour cela le tutoriel mentionné ci-dessous et l'aide en ligne dans OpenOffice.org) ni à l'API d'OpenOffice.org (voir à ce sujet la Référence de l'API et le Guide du Développeur, cités ci-dessous également.)

Le but de ce document est de vous faire comprendre l'application de macros dans votre utilisation d'OpenOffice.org.

Un **enregistreur de macro** est en cours de développement (NdT: il est déjà disponible dans la version de développement d'OpenOffice.org 643) et sera rajouté à OOO dans un avenir proche. Ceci devrait simplifier la création et l'utilisation de macros de manière significative.

Une **macro** est un morceau de code en langage BASIC qui a pour but d'exécuter une tâche précise dans OOO.

Par exemple :

```
Sub afficher_un_message

    MsgBox "Ceci est un message concernant ...", 16, "Display
Message"

End Sub
```

Lorsque l'on exécute cette sous-routine, elle affichera une première chaîne de caractères dans une boîte de dialogue, avec la deuxième chaîne de caractères sous forme de titre dans la barre de titres.

Autre exemple

```
Function prix_soldé ( prix_de_vente As Double ) As Double

    prix_soldé = 0.80 * prix_de_vente
```

End Function

Cette fonction multiplie la valeur `prix_de_vente`, qui lui est donné en paramètre, par un facteur de multiplication (en l'occurrence 0.80) et renvoie un `prix_soldé` correspondant.

Des macros peuvent être regroupées ensemble dans des Modules, et des Modules peuvent être regroupés en Bibliothèques.

Des macros peuvent être exécutées par un utilisateur final, ou appelées en cliquant sur un bouton.

Elles peuvent être déclenchées par une modification dans un contrôle d'interface utilisateur, telle que, par exemple, une sélection faite par un utilisateur dans une liste de choix ou en cliquant sur un bouton radio ou en cochant une case.

Elles peuvent être appelées lors d'événements prédéfinis, par exemple, lors de l'ouverture d'un document.

Elles peuvent être appelées par une combinaison de touches clavier.

Elles peuvent également être appelées en cliquant sur un bouton dans une barre d'outils.

Elles peuvent être appelées en cliquant sur un lien hypertexte.

Parfois, elles sont aussi appelées par d'autres macros.

Des macros exprimant une fonction peuvent être appelées en tant que formule dans des feuilles de calcul.

Elles peuvent également être impliquées dans la validation de données de cellules.

Elles peuvent être appelées depuis des boîtes de dialogue qui est elle-même créée par une ou plusieurs macros.

Comment Trouver des Informations sur les Macros

Bien que la documentation concernant des macros et l'API d'OpenOffice.org (interface de programmation d'applications) ait été difficile à dénicher, la situation s'améliore de jour en jour.

Le Star BASIC Programming Tutorial, qui couvre l'ancien code Star BASIC 5 se trouve à l'adresse suivante :

<http://api.openoffice.org/basic/man/tutorial/tutorial.pdf>

Le projet OOO API se trouvant à <http://api.openoffice.org/> comprend des liens vers le nouveau Kit de Développement Logiciel (SDK), qui contient la référence API, et vers le Guide du Développeur, qui lui contient de petits exemples.

Vous pouvez également chercher dans l'archive de la liste de discussion dev@api à l'adresse suivante : <http://api.openoffice.org/servlets/SummarizeList?listName=dev>

dans l'archive de la liste dev@dba à l'adresse :

<http://dba.openoffice.org/servlets/SummarizeList?listName=dev>

dans l'archive de la liste users@openoffice.org à l'adresse :

<http://www.openoffice.org/servlets/SummarizeList?listName=users>

et le projet de Kit de Développement UNO à l'adresse : <http://udk.openoffice.org/>

Des exemples de code de macros peuvent être téléchargés depuis les sections de OooDocs.org à l'adresse :

<http://www.ooodocs.org/modules.php?name=Forums&file=index>

notamment dans les sections Macros and API, Tips & Tricks, Data Sources, Calc, and Writer

et également à OoExtras.org: <http://www.ooextras.org/>

et aussi à Darwinwars: http://www.darwinwars.com/lunatic/bugs/oo_macros.html

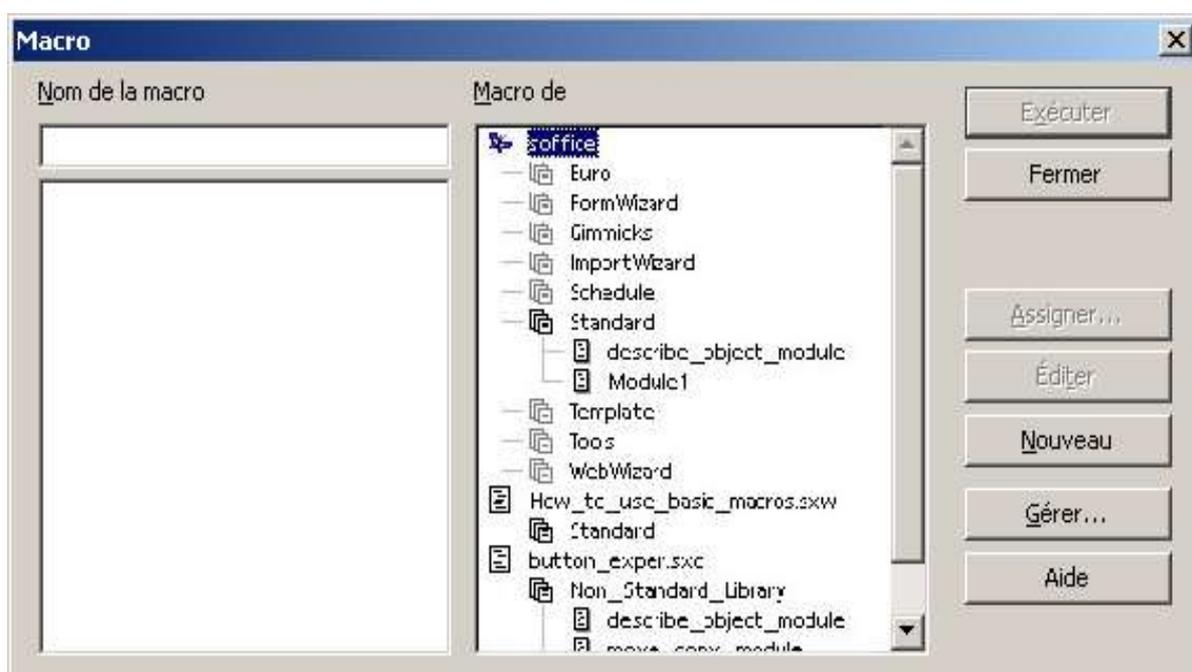
et encore à : <http://api.openoffice.org/unbranded-source/browse/~checkout~/api/odk/examples/examples.html>

Utilisation de la Boîte de Dialogue Macro

Si vous voulez travailler avec des macros, ouvrez la boîte de dialogue Macro :

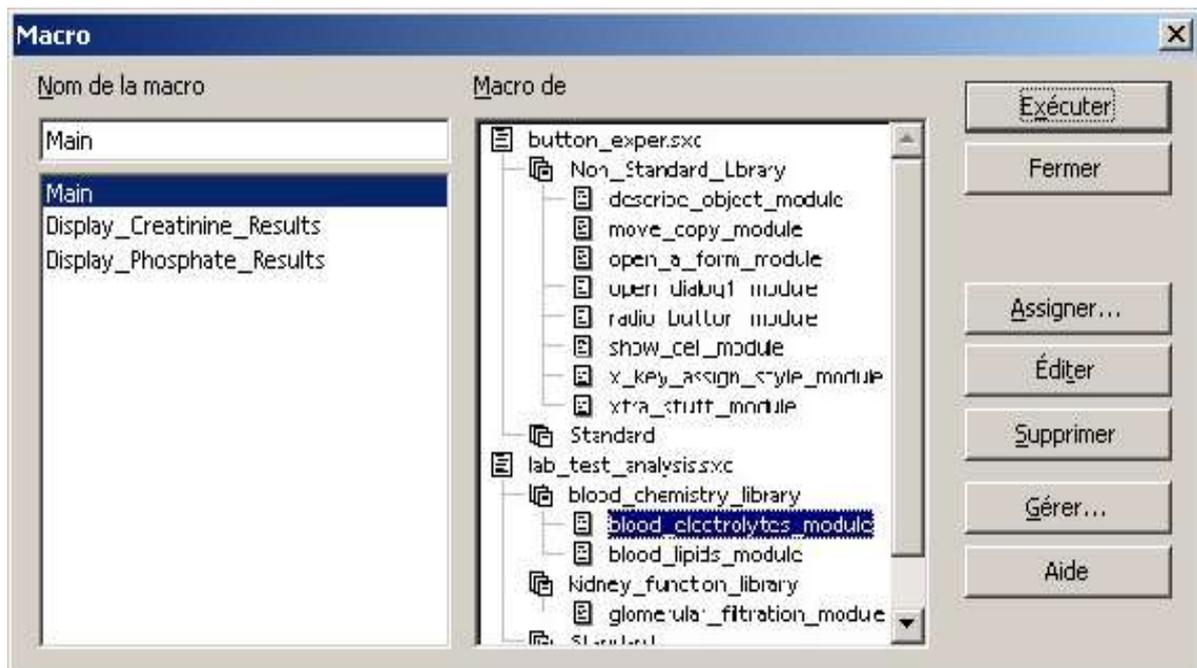
Choisir **Outils> Macro** dans le menu principal.

Dans la fenêtre de droite, intitulée **Macro de:** vous verrez une liste de documents, de bibliothèques, et de modules contenant diverses macros. En haut, sous la rubrique **soffice**, se trouvent plusieurs bibliothèques qui sont partagées et disponibles pour tout document OpenOffice.org, et comprenant la bibliothèque désignée Standard et le module designé Module1.



Par exemple, le **describe_object_module** dans la bibliothèque **Standard** sous **soffice** contient des macros qui sont disponibles pour tout document OOo sur cet ordinateur précis.

Vers le bas de la fenêtre de droite (faire défiler) se trouvent les noms de tous les documents qui sont à présent ouverts, ainsi que les bibliothèques et les modules qu'ils contiennent.

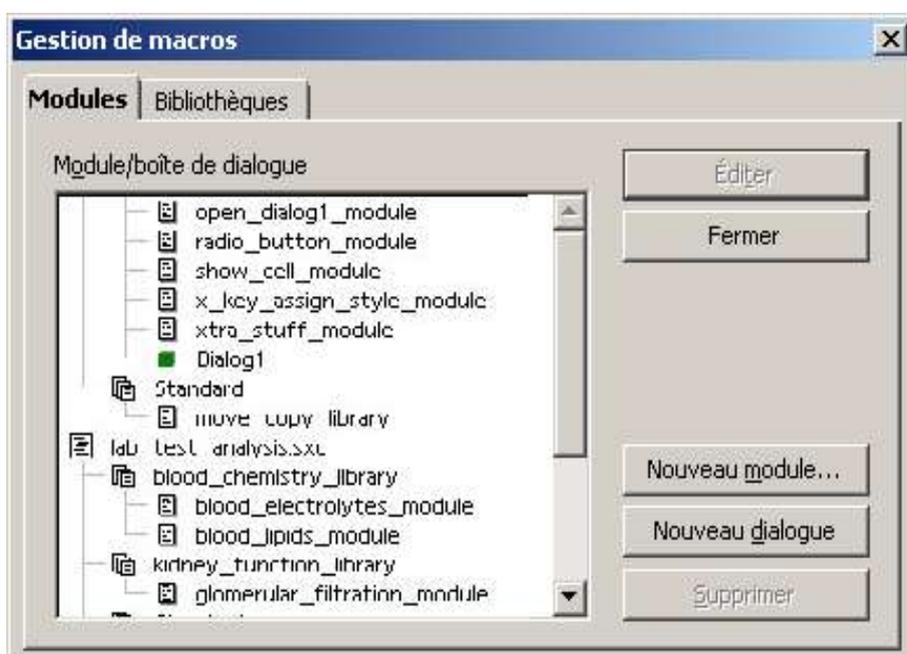


La figure montre deux feuilles de calcul, `button_exper.sxc` et `lab_test_analysis.sxc`, ainsi que leurs bibliothèques et modules associés. La fenêtre de gauche indique les noms des trois sous-routines de macro dans le module `blood_electrolytes_module`, qui est mis en surbrillance.

Nommer une bibliothèque `blood_chemistry_library` ou un module `blood_lipids_module` rend la compréhension de leur fonctionnement plus clair que si l'on utilise les noms bibliothèque `Standard` et `Module1`.

Utilisation du Gestionnaire de Macros

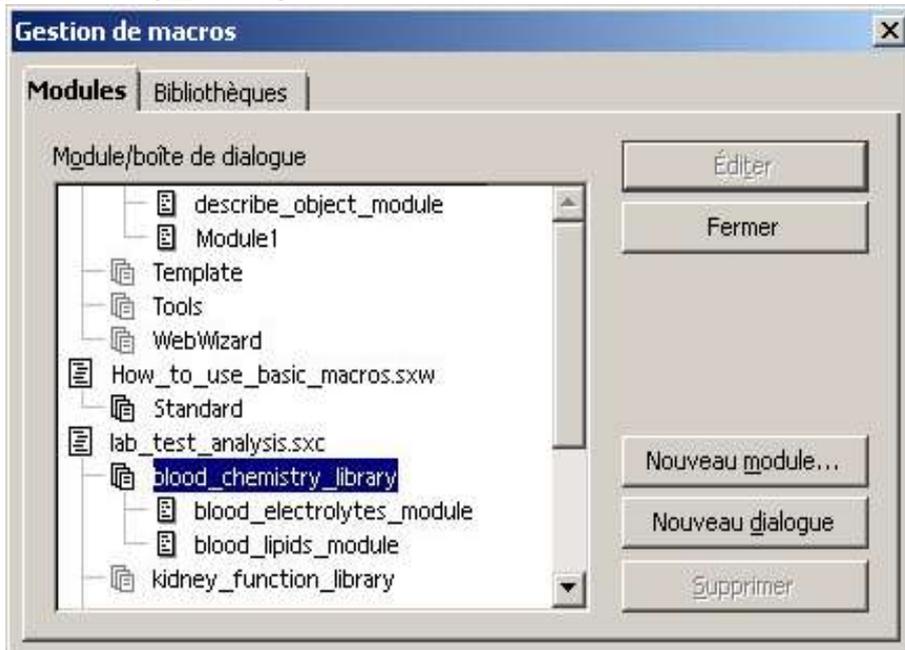
Lorsque vous cliquez sur le bouton Gérer dans la boîte de dialogue des macros, la fenêtre du Gestionnaire de Macros s'ouvrira, ce qui vous permettra de créer de nouvelles bibliothèques, de nouveaux modules, et de nouvelles boîtes de dialogue.



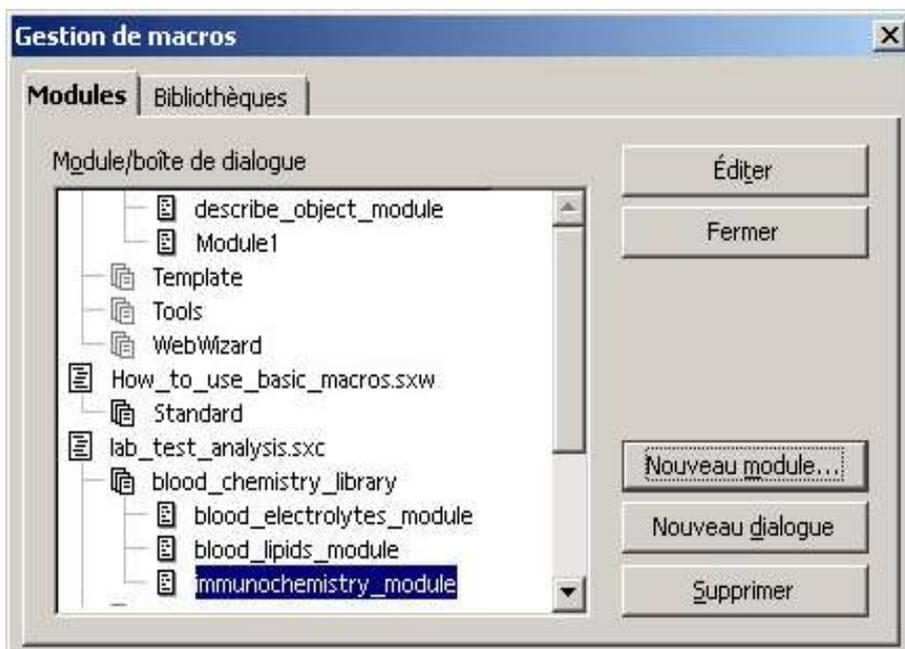
Création d'un Nouveau Module

L'onglet **Modules** est choisi par défaut. Vous pouvez choisir une bibliothèque existante pour stocker votre nouveau module.

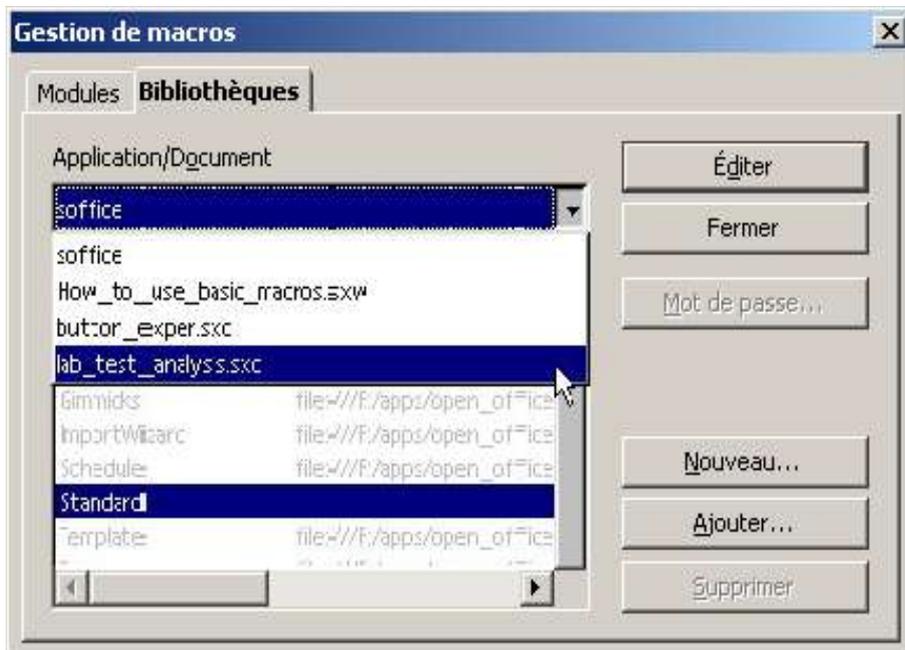
Dans l'exemple donné ci-après, nous avons sélectionné la bibliothèque **blood_chemistry_library**.



Et maintenant, ajoutons-y un nouveau module en cliquant sur le bouton Nouveau Module.

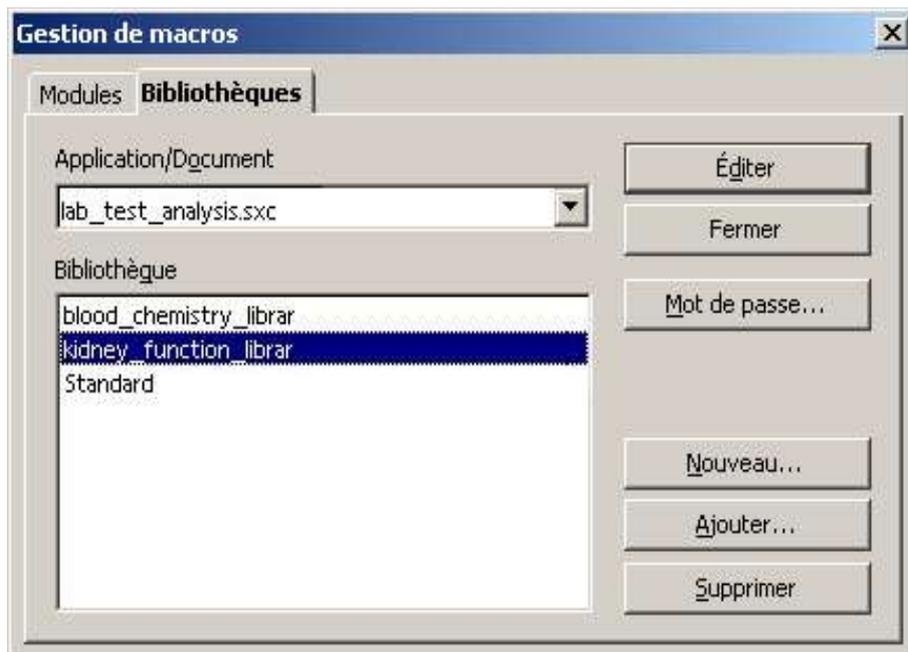


Création d'une Nouvelle Bibliothèque



Cliquez sur l'onglet **Bibliothèques** dans le Gestionnaire de Macros. Le menu déroulant supérieur, intitulé **Application/Document**, vous permet de sélectionner un document en particulier dans lequel vous allez créer votre nouvelle bibliothèque. Ici, nous avons choisi la feuille de calcul **lab_test_analys.sxc**.

Après avoir sélectionné un document, vous pouvez créer une nouvelle bibliothèque pour ce document ou bien éditer une bibliothèque existante. Ici, nous cliquerions sur le bouton **Nouveau...** afin d'entrer le nom de notre nouvelle bibliothèque pour ce document.



Vous pouvez en apprendre plus sur les bibliothèques de macro en lisant l'aide en ligne fournie avec OOo. Dans le menu, choisissez

Aide > Sommaire > Contenu et double-cliquez sur

[Aide pour OpenOffice.org Basic](#) puis sur

[Programmation avec OpenOffice.org Basic](#) et puis sur [Technical Data](#).

Lire également [Integrated Development Environment \(IDE\)](#) ainsi que la rubrique dépendante

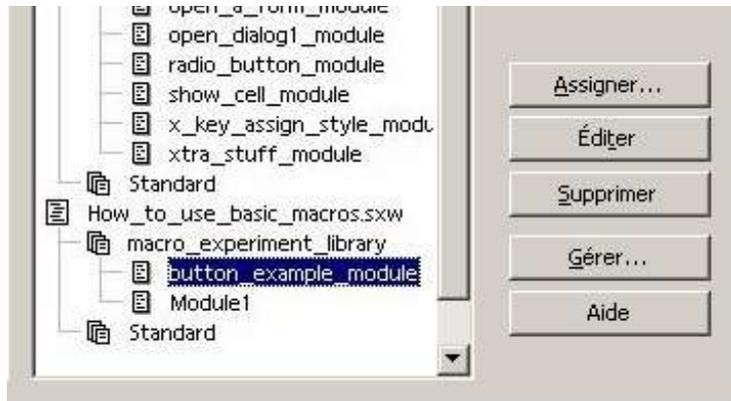
[Création et Organisation de Bibliothèques et Modules](#) et la rubrique [Options Supplémentaires](#).

A lire également, la rubrique [Run-Time Functions](#), qui décrit les fonctions de programmation et les opérateurs BASIC disponibles.

Ecrire une Macro

Ecrivons maintenant une macro pour ouvrir un autre document OOo.

A partir du menu principal, choisissez **Outils > Macro** pour ouvrir la boîte de dialogue de macros.



Ici, nous avons déjà ajouté une nouvelles bibliothèque, intitulée **macro_experiment_library**, et un nouveau module, intitulé **button_example_module**, au document courant.

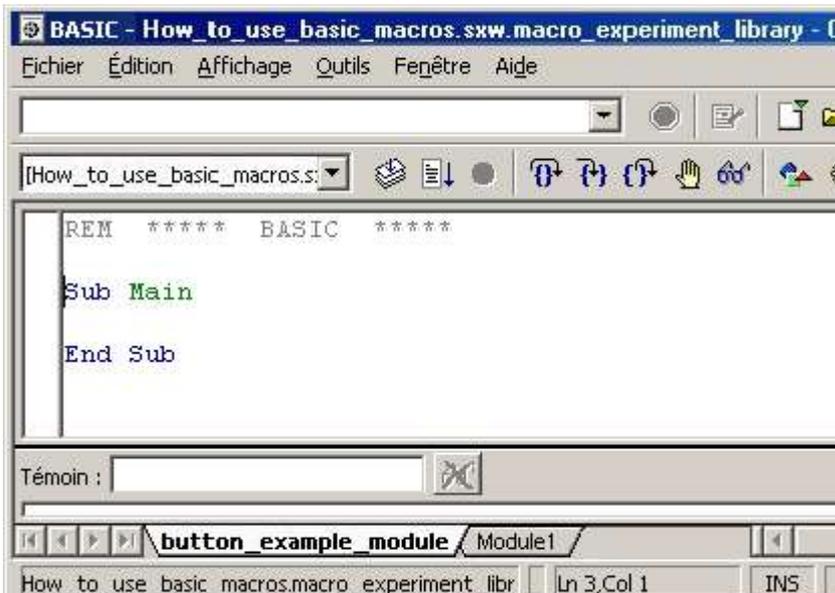
Nous avons décidé de placer notre macro dans le module **button_example_module**.

Lorsque le module est sélectionné, cliquez sur le bouton **Éditer** dans la boîte de dialogue de Macros afin d'ouvrir l'**Environnement de Développement Intégré ou BASIC EDI**.

L'EDI est une fenêtre d'édition de macros.

Voici l'EDI montrant un nouveau module « vide ».

Remarquez le nom **button_example_module** sur l'onglet en bas.



```
REM ***** BASIC *****  
  
Sub Main  
  
End Sub
```

The screenshot shows a window titled "BASIC - How_to_use_basic_macros.sxw.macro_experiment_library - O". The menu bar includes "Fichier", "Édition", "Affichage", "Outils", "Fenêtre", and "Aide". The toolbar contains various icons for editing and navigation. The main text area displays the BASIC code for a macro. Below the code is a "Témoignage" (Witness) field. At the bottom, the status bar shows the current file path, line number (Ln 3), column number (Col 1), and the current keyboard state (INS).

Il existe déjà une sous-routine intitulée **Main** .

Dans une macro, toute ligne ou partie d'une ligne commençant avec une **apostrophe** ou avec **REM** est une ligne de **commentaire**. Les commentaires sont destinés à faciliter la compréhension et la lecture du code de programmation par l'utilisateur et sont ignorés par l'interpréteur du langage BASIC.

Supprimer la routine Main. Tapez, ou copiez-collez la macro suivante.

Changez le chemin et le nom pour correspondre à un document OoO sur votre machine.

```
' -----  
  
Sub ouvrir_un_formulaire  
  
' ce qui suit va ouvrir une copie séparée d'un document existant  
  
  Dim mNoArgs()  
  Dim oDesktop As Object  
  Dim oDocument As Object  
  Dim sUrl As String  
  
  oDesktop = createUnoService("com.sun.star.frame.Desktop")  
  
  sUrl = "file:///E:/users/exper_dbase/Form_exper_dbase.sxw"  
  
  oDocument =  
    oDesktop.LoadComponentFromURL(sUrl, "_blank", 0, mNoArgs())  
End Sub  
  
' -----
```

Lorsqu'on exécute cette macro, elle ouvrira le document OpenOffice.org Writer intitulé Form_exper_dbase.sxw qui se trouve sur un lecteur E: (Windows) dans le répertoire /users/exper_dbase (en supposant que le document existe bien entendu).

Sur une machine Linux / Unix, le format de la commande serait :

```
sUrl = "file:///home/exper_dbase/Form_exper_dbase.sxw"
```

Pour ouvrir un nouveau document OpenOffice.org Writer vide, utilisez:

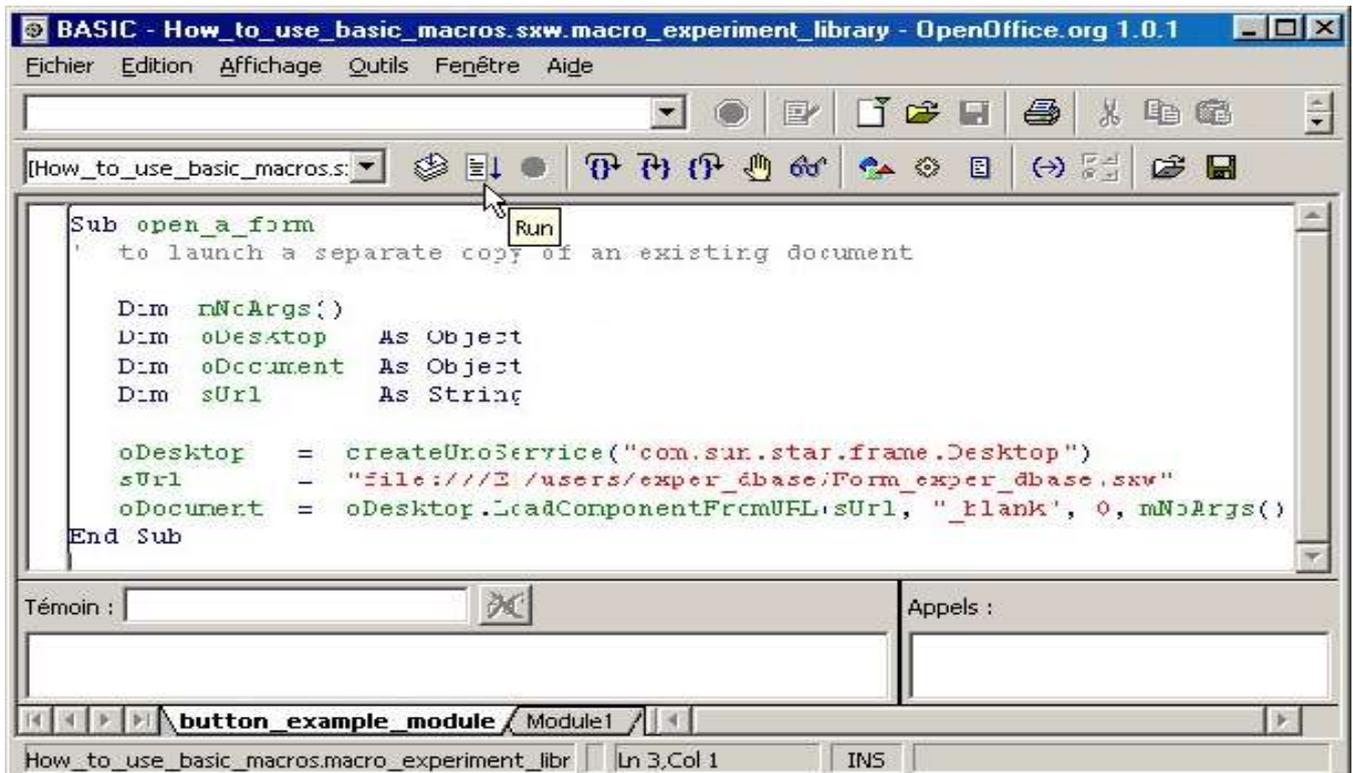
```
sUrl = "private:factory/swriter"
```

Pour ouvrir une nouvelle feuille de calcul OpenOffice.org Calc, utilisez :

```
sUrl = "private:factory/scalc"
```

Exécution d'une Macro au sein de l'EDI

Pour exécuter la macro `ouvrir_un_formulaire`, cliquez sur le bouton **Exécuter**.



Si le document existe, il devrait s'ouvrir dans une nouvelle fenêtre OOo.

La plupart du temps, on ne fait exécuter une macro dans l'EDI que pour la déboguer et la tester. Il est souvent plus aisé de la lancer par d'autres moyens.

Dans l'aide **Aide > Sommaire > Index**, lisez l'article sur [fonction bars; macros](#).

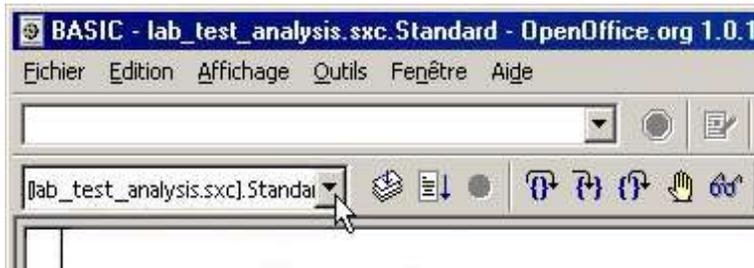
Vous pouvez également faire exécuter une macro depuis la boîte de dialogue de macro, en la sélectionnant et en cliquant sur le bouton **Exécuter**.

Vous pouvez aussi démarrer OOo avec une macro depuis la ligne de commande, par exemple :

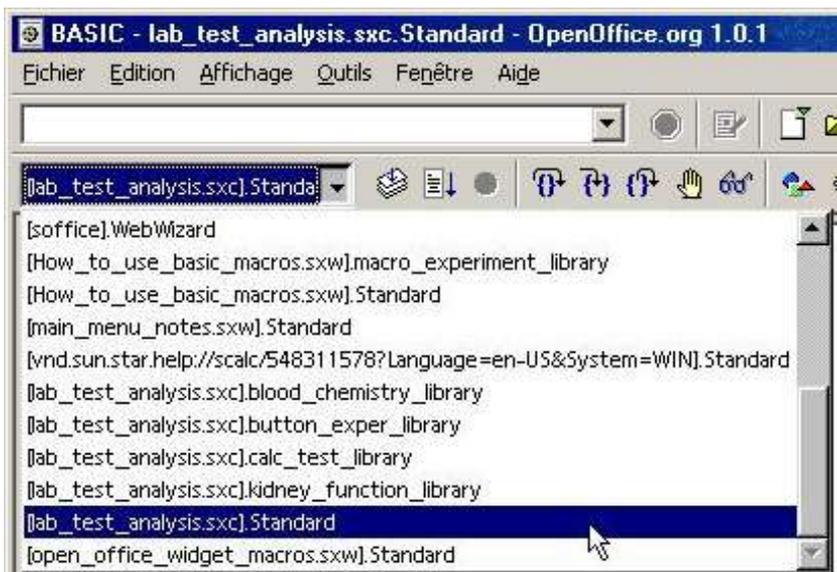
```
soffice macro:///some_global_library.some_module.some_method
```

Sélectionner des Bibliothèques dans l'EDI

Dans l'EDI, vous pouvez changer une bibliothèque pour une autre se trouvant dans le conteneur Soffice ou dans tout document actuellement ouvert. Utilisez pour ceci la liste déroulante sur l'extrémité gauche de la deuxième barre d'outils :

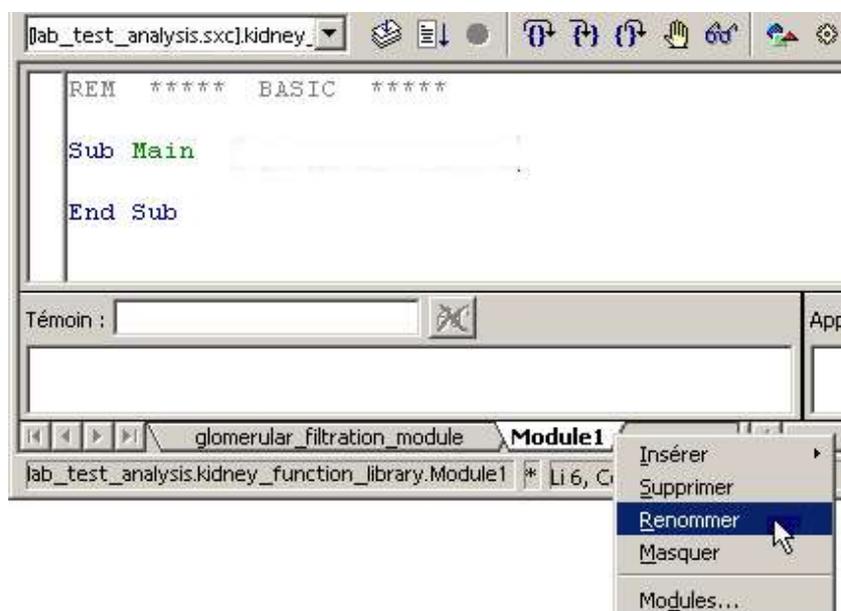


Cliquez pour afficher la liste de bibliothèques disponibles.



Sélectionner la bibliothèque que vos voulez éditer et tous ses modules apparaîtront sous forme de **pages à onglet** dans l'EDI, avec les onglets en bas.

Vous pouvez renommer un module en cliquant sur son onglet dans l'EDI, puis avec le bouton droit de la souris, en choisissant **Renommer**.



Ici, nous nous apprêtons à renommer **Module1** dans la bibliothèque **kidney_function_library**.

Appel d'une Macro Depuis une Autre Macro

Un des moyens les plus simples pour appeler une macro ou une fonction est de l'utiliser dans une autre macro ou fonction.

```
Sub Main
    afficher_qqchose_utile
End Sub

Sub afficher_qqchose_utile
    MsgBox "La surface d'un cercle de rayon 2,36 est: " + _
        cStr(surface_du_cercle(2.36)), 64, "Surface"
End Sub

Function surface_du_cercle ( radius As Double ) As Double
    surface_du_cercle = 3.14159 * radius * radius
End Function
```

(**cStr** convertit un nombre en une chaîne de caractères afin qu'il puisse être affiché en tant que texte.)

Cliquer sur le bouton pour voir l'effet de ce jeu de macros :

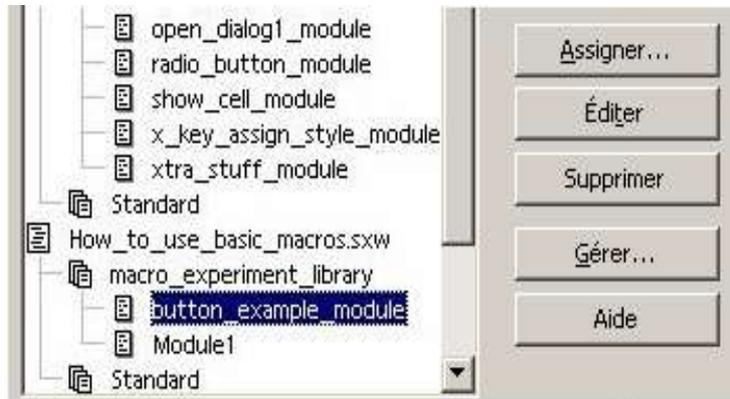


L'exemple du bouton est expliqué ci-après.

Appeler une Macro avec un Bouton

Vous pouvez appeler une macro en la liant à un bouton.

Choisissez **Outils > Macro** afin d'ouvrir la boîte de dialogue de macro.



Sélectionnez **button_example_module** et cliquez **Éditer** pour ouvrir l'EDI BASIC.

Nous allons saisir une petite macro pour afficher un message. Ensuite, nous lierons cette macro à un bouton.

```
Sub afficher_un_message
    MsgBox "Hello, world!", 16, "Afficher le Message Hello"
End Sub
```

Afin de créer notre propre bouton, cliquez et maintenez la souris sur le bouton **Formulaire** dans la barre d'outils de gauche.



La barre d'outils de formulaire s'affichera. Cliquez sur le bouton **Bouton**.

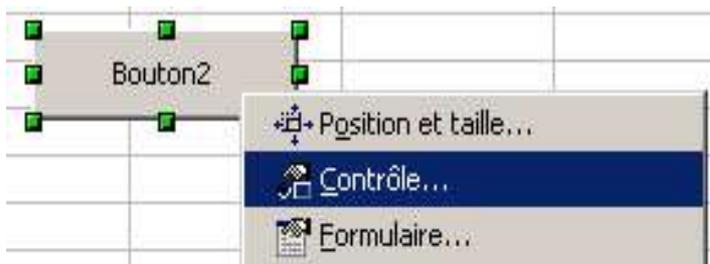


Sur la page, tirer la silhouette rectangulaire du premier Bouton, jusqu'à la taille voulue, et relâchez la souris.

Vous devriez voir affiché un bouton gris rectangulaire intitulé BoutonX muni sur son pourtour de 8 petits carrés verts appelés « poignées ».

Les poignées indiquent que le bouton se trouve en **Mode de Conception de Formulaire**, dans lequel vous pouvez modifier ses taille, position et autres propriétés. Dans ce mode, le bouton ne réagit pas comme un bouton, mais comme un élément ou composant de formulaire. Plus tard, vous quitterez ce mode et entrerez en **Mode d'Exécution de Formulaire**, dans lequel le bouton agira en tant que bouton.

Cliquez maintenant avec le bouton droit de la souris pour afficher le menu contextuel et choisissez **Control...**



Ceci ouvrira la boîte de dialogue des Propriétés du bouton, où il vous est possible de changer plusieurs propriétés du bouton, telles que son nom, son étiquette, sa couleur, etc.

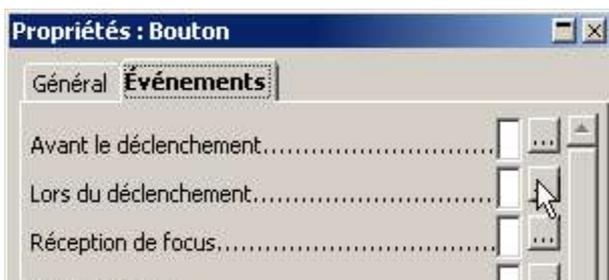
Boîte de dialogue des Propriétés:



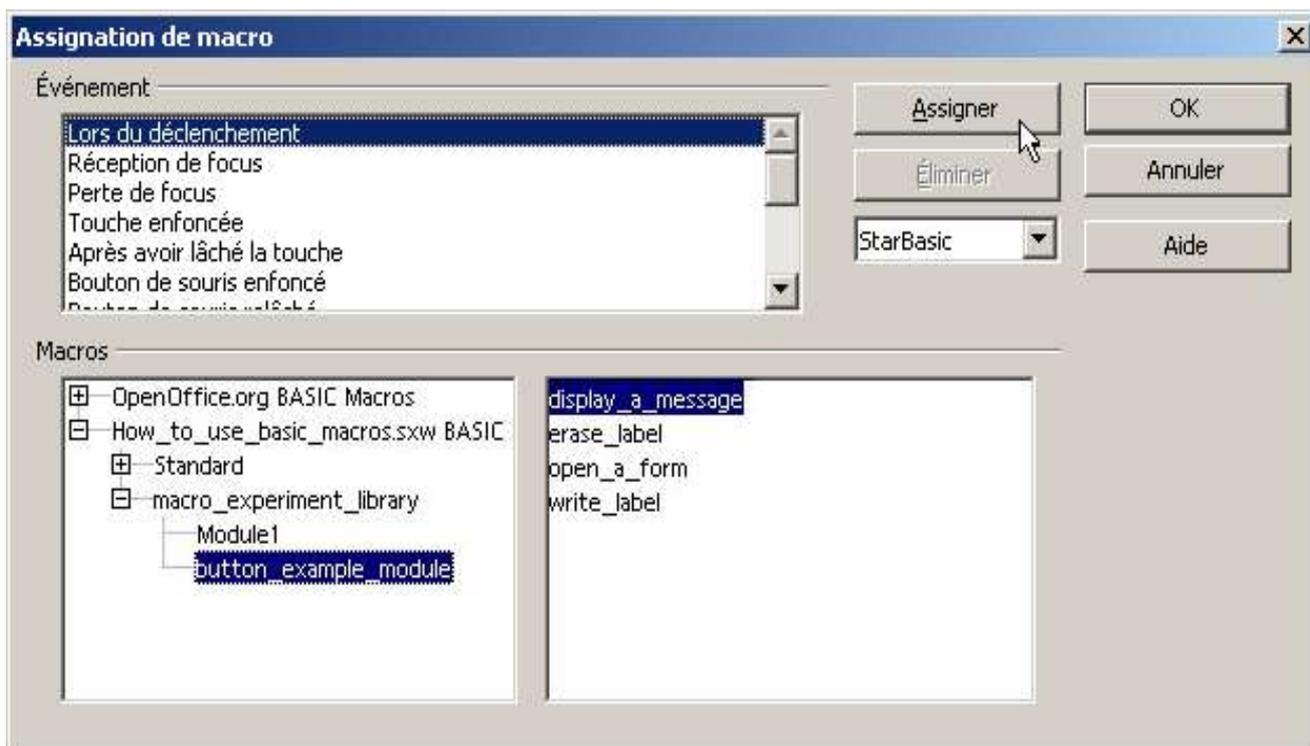
Changez la propriété **Label** pour **Open a Document**. Changez la couleur d'arrière-plan **Background color** pour **Green 2**. Changez le jeu de caractères **Character set** pour **Comic Sans MS 12pt orange**.

Bien entendu, vous pouvez choisir ces choses-là vous-mêmes.

Cliquez maintenant sur l'onglet **Events**. Cliquez sur le bouton correspondant à l'événement **When initiating**.



Ceci aura pour effet d'ouvrir la boîte de dialogue pour attribuer une macro à un événement.



Dans les deux fenêtres inférieures, nous avons sélectionné notre module et la macro **afficher_un_message** et sommes maintenant prêt à cliquer sur **Assigner** pour lier cette macro au bouton.

Cliquez sur **Assigner** et puis cliquez sur **OK** pour fermer la boîte de dialogue des Propriétés.

Cliquer sur le bouton **Formulaire** pour quitter le mode de conception de formulaires et pour retourner dans le mode d'exécution de formulaires.

Le bouton que vous venez de créer devrait désormais fonctionner normalement.

Cliquez sur le bouton **Display a Message** pour l'essayer.

A rectangular button with a light gray background and a thin border. The text "Display a Message" is centered on the button in an orange, sans-serif font.

Vous verrez le message apparaître dans une boîte d'affichage de message MsgBox :

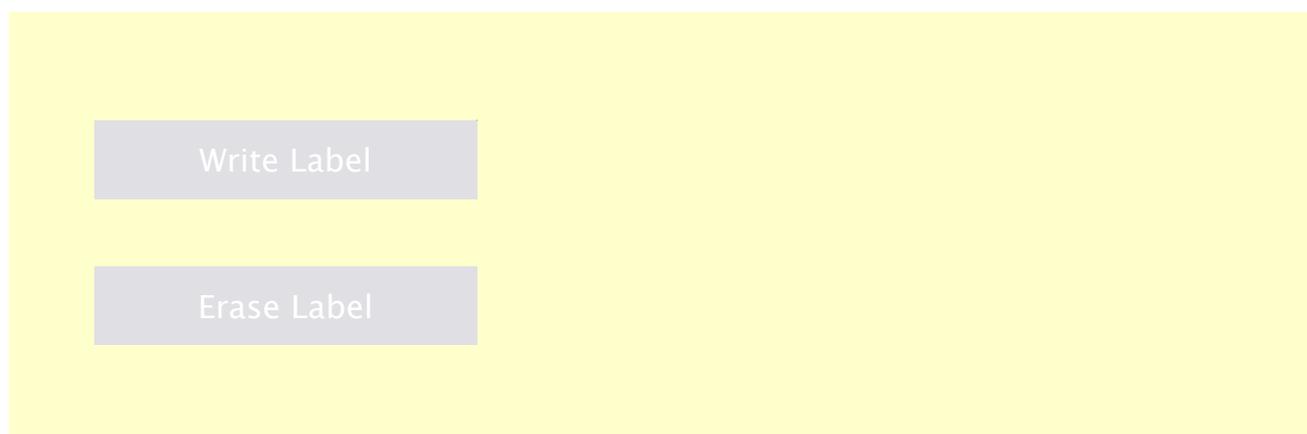


Cliquez sur OK pour fermer le message.

Astuce : vous pouvez atteindre la boîte de dialogue de configuration pour attribuer une macro par le biais de **Outils > Macro** en sélectionnant le nom d'une macro existante et en cliquant sur le bouton **Assigner....**

Exemple avec Deux Boutons et une Etiquette

L'un des deux boutons écrit un message dans l'étiquette verte, l'autre l'efface. Essayez en cliquant sur le bouton bleu d'abord.



Essayons maintenant de reproduire cet exemple.

Nous devons d'abord écrire deux macros.

Choisissez **Outils > Macro** pour ouvrir la boîte de dialogue de macro.

Sélectionnez **button_example_module** et cliquez sur **Éditer** pour ouvrir l'EDI

BASIC.

Tapez ou copiez-collez les deux macros suivantes.

```
' -----  
  
Sub écriture_étiquette  
    Dim oDocument      As Object  
    Dim oForm           As Object  
    Dim oLabel          As Object  
  
    oDocument          = ThisComponent  
    oForm              = _  
        oDocument.DrawPage.Forms.getByName  
("écrire_supprimer_formulaire")  
    oLabel             = oForm.GetByName  
( "écrire_supprimer_étiquette" )  
    oLabel.Label      = "OpenOffice.org office suite"  
End Sub  
  
' -----  
  
Sub erase_label  
    Dim oDocument      As Object  
    Dim oForm           As Object  
    Dim oLabel          As Object  
  
    oDocument          = ThisComponent  
    oForm              = _  
        oDocument.DrawPage.Forms.getByName  
("écrire_supprimer_formulaire")  
    oLabel             = oForm.GetByName( "écrire_supprimer_étiquette"  
)  
    oLabel.Label      = ""  
End Sub  
  
' -----
```

Remarquez que les lignes qui sont trop longues doivent être interrompues avec le caractère tiret bas '_' en tant que caractère de suite pour continuer sur la ligne suivante, tel que oForm = _

Après avoir écrit les macros, vous pouvez dessiner vos boutons et l'étiquette sur la page.

Pour créer ces boutons et l'étiquette, cliquez et maintenez le bouton **Formulaire** dans la barre d'outils la plus à gauche de votre écran. La barre d'outils de formulaire apparaîtra. Sélectionnez le bouton **Bouton**.

Sur la page, dessinez votre premier Bouton. Relâchez la souris lorsque le bouton a la taille voulue.

Cliquez, avec le bouton droit de la souris, sur le bouton que vous venez de dessiner pour afficher le menu contextuel et sélectionnez **Contrôle...** pour ouvrir les propriétés du bouton.



Changez la propriété **Titre** en **Write Label**. Changez la propriété de couleur d'arrière plan **Couleur d'Arrière plan** en **Bleu**. Changez le jeu de caractères **Jeu de caractères** en **Arial 12pt white**.

Maintenant cliquez sur l'onglet **Evénements**. Cliquez sur le bouton d'événement **Lors du déclenchement**.

Ceci ouvrira la boîte de dialogue permettant d'attribuer des macros à des événements.

Sélectionnez le module et la macro **write_label** et cliquez sur **Assigner** pour lier cette macro au premier bouton, puis cliquez sur **OK**.

Fermez la boîte de dialogue.

Répétez les étapes ci-dessus pour dessiner votre deuxième bouton, avec les différences suivantes :

Changez la propriété **Titre** en **Erase Label**. Changez la propriété de couleur d'arrière plan **Couleur d'arrière plan** en **Magenta**. Changez le jeu de caractères **Jeu de caractères** en **Arial 12pt white**.

Fermez la boîte de dialogue des propriétés du bouton.

Après avoir cliqué sur l'onglet **Événements**, puis **Lors du déclenchement**, sélectionnez le module et la macro **erase_label** et puis cliquez sur **Assigner** pour lier cette macro au deuxième bouton.

Créons maintenant l'étiquette.

Cliquez sur le bouton **Formulaires** pour faire apparaître la barre d'outils complète et cliquez ensuite sur le bouton **Champ d'étiquette**.



Dessinez la forme de l'étiquette comme vous l'aviez fait pour les deux boutons.

Cliquez du bouton droit de la souris sur l'étiquette, choisissez **Contrôle ...**

Changez la propriété **Nom** en **écrire_supprimer_étiquette**. Changez la couleur d'arrière plan **Couleur d'arrière plan** et le jeu de caractères **Jeu de caractère** en vos choix personnels. Mettre l'**Alignement** sur **Centrer**.

Fermez la boîte de dialogue.

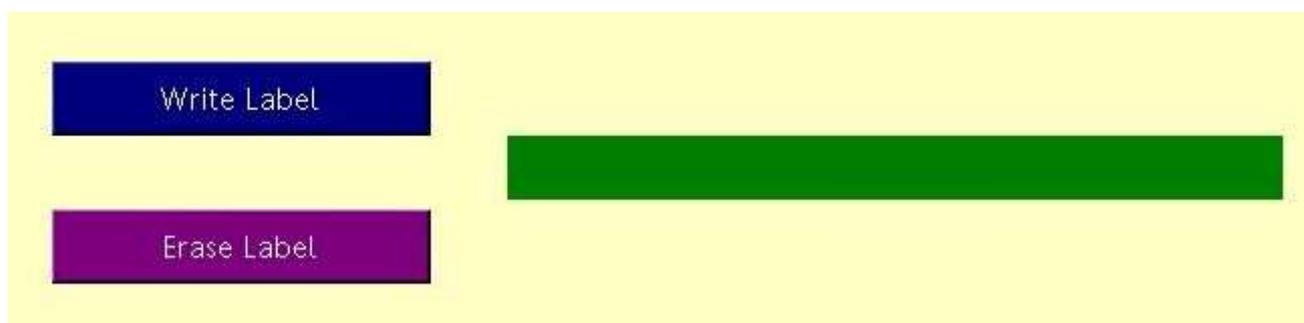
Cliquez sur le bouton **Formulaires** pour sortir du mode conception.

L'exemple décrit ci-dessus devrait maintenant fonctionner.

En cliquant sur l'étiquette bleu **Write Label** vous devriez voir apparaître :



En cliquant sur bouton magenta **Erase Label** , vous ferez disparaître ce que vous venez d'écrire.

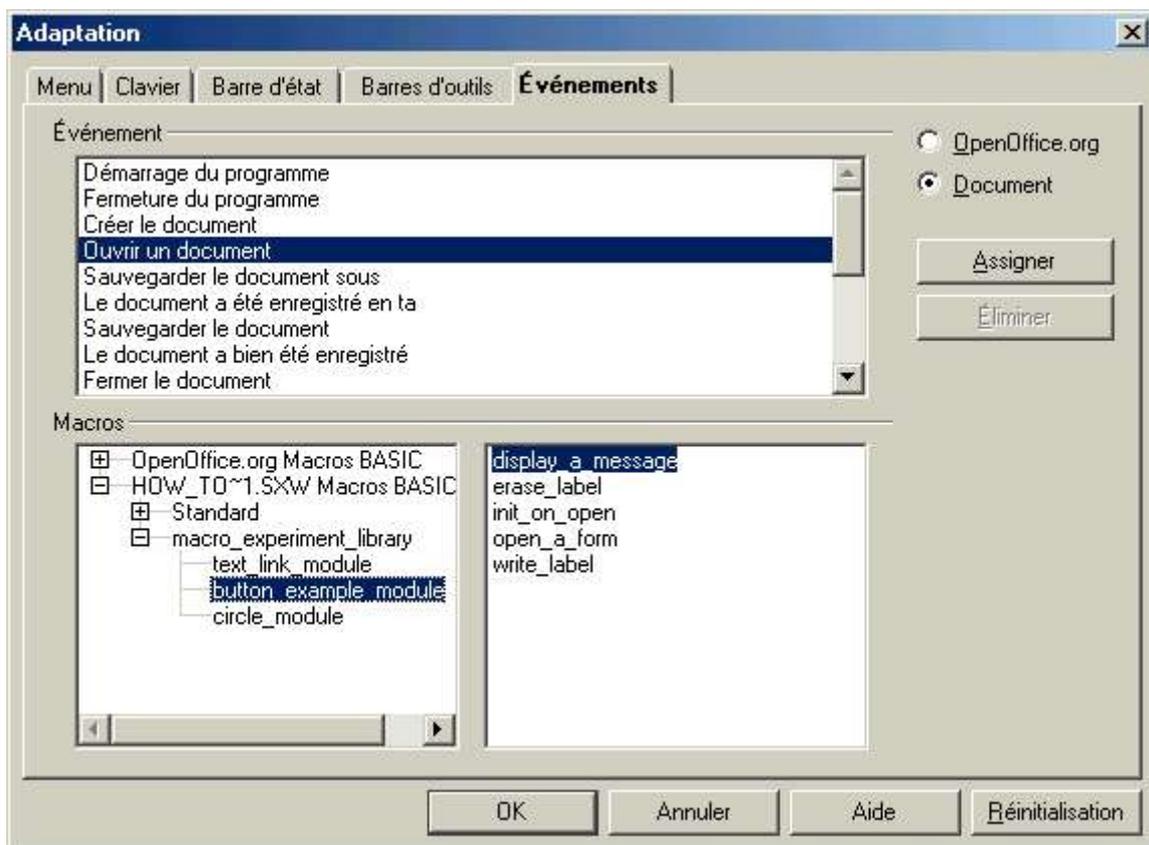


Allons jeter un oeil à d'autres façons de lancer des macros.

Appeler une macro Depuis un Événement

Il existe **17 événements**, prédéfinis, tels que la création, l'ouverture, l'enregistrement, ou la fermeture d'un document, qui peuvent déclencher une macro à chaque fois qu'un tel événement a lieu.

Dans le menu principal, choisissez l'onglet **Outils > Adaptation > Événements** afin d'ouvrir la boîte de dialogue de **Configuration**.



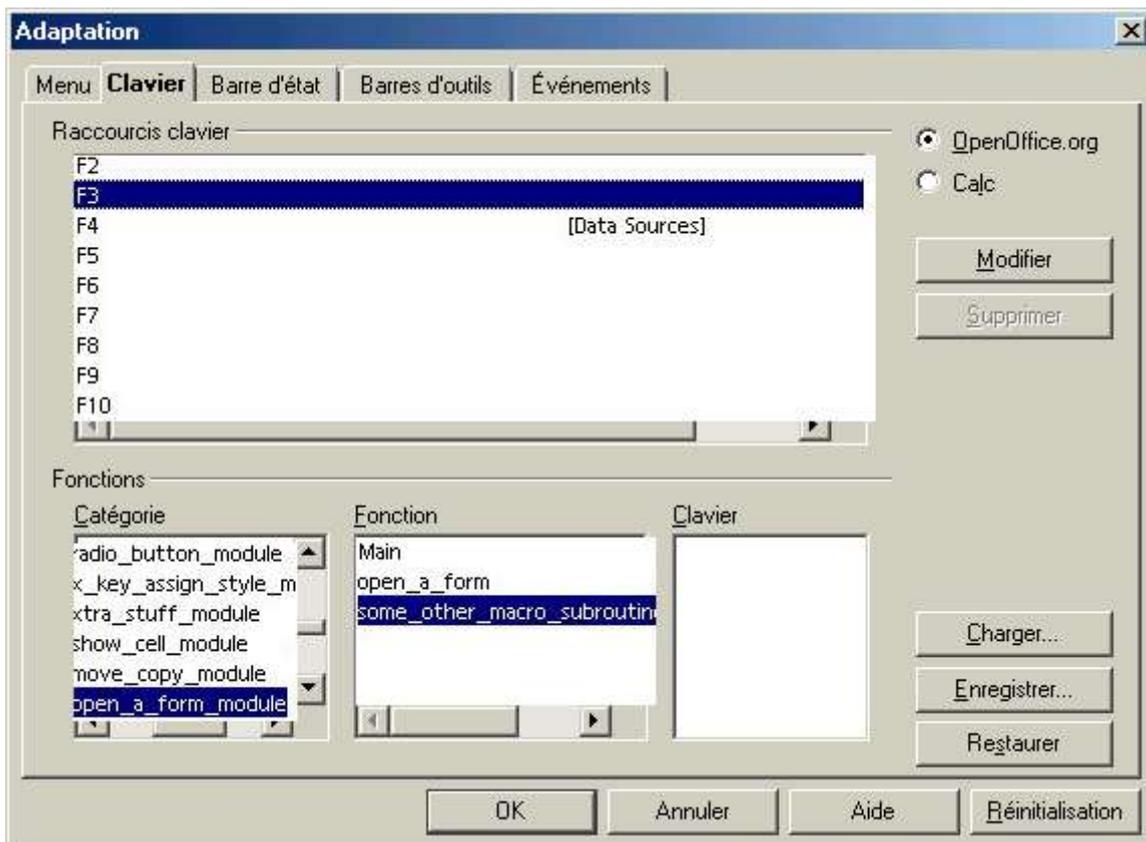
Ici, nous allons attribuer la macro **init_on_open** à l'événement **Ouverture de Document**. La macro sera lancée à la fin de l'ouverture du document.

Ouvrez la boîte de dialogue **Configuration** à nouveau et regardons les autres onglets. Vous pouvez aussi attribuer une macro à une entrée de **menu** nouvelle ou existante ou à une combinaison de touches **clavier**.

Appeler une Macro depuis le Clavier

Une macro peut être attribuée à une combinaison de touches clavier. Dans le menu principal, choisissez :

Outils > Adaptation > Onglet Clavier afin d'ouvrir la boîte de dialogue **Configuration**. Sélectionnez l'un des boutons radio OpenOffice.org ou Calc/Writer selon que la touche clavier est censée faire exécuter la macro dans tous les documents OoO ou dans un type de documents seulement.



Regarder la liste des **raccourcis clavier** pour trouver une touche ou une combinaison de touches qui vous semble correcte et qui soit disponible. Il existe un certain nombre de touches simples, et des combinaisons Maj+touche, Ctrl+touche, and Ctrl+Maj+touche. Pour l'instant, il n'existe aucune combinaison de touches Alt+touche.

Dans la fenêtre **Catégorie**, faites dérouler jusqu'en bas de la liste et trouvez les documents, bibliothèque et modules contenant la macro que vous voulez lier à la touche. Lorsque vous sélectionnez le module, les noms de macros s'y trouvant apparaîtront dans la fenêtre **Fonction**. Dans la figure, nous nous apprêtons à lier la macro **some_other_macro_subroutine** à la touche **F3**, non encore utilisée. Cliquez

sur **Modifier** pour attribuer la macro à la touche. La touche F3 apparaîtra dans la fenêtre **Clavier** .

Fermez la boîte de dialogue de Configuration.

Si vous appuyez maintenant sur la touche F3, vous lancerez la macro **some_other_macro_subroutine** .

Une attribution de combinaison de touches reste valable tant que vous ne la désaffectez pas, peu importe le type de document ouvert. Toutefois, si la macro est liée à un document spécifique, vous ne pourrez la lancer sauf si ce document-là est ouvert. Si vous avez une macro très utile que vous voulez attribuer à une combinaison de touches, il vaut mieux la placer dans le jeu **soffice** de bibliothèques de manière à ce qu'elle soit toujours disponible. Si la macro n'a qu'une utilité pour un seul document, elle peut être placée dans ce document. Si elle est utile pour une classe ou un groupe de documents semblables, mettez la dans un modèle de document qui servira à créer ces documents. Si la macro n'est pas disponible, la combinaison de touches n'a aucun effet.

Vous pouvez en apprendre plus sur le clavier en lisant l'aide en ligne. Dans le menu principal, choisissez :

Aide > Sommaire > Index et tapez le mot-clé **clavier**. Consultez les articles **clavier; configuration** et **clavier; personnalisation**.

Vous trouverez d'autres renseignements sur les menus dans l'aide en ligne. Utilisez le mot clé **menus**. Consultez les articles **menus; configuration, exemples** et **menus; personnalisation**.

Appeler une Macro avec un Bouton de Barre d'Outils

Il vous est possible de lier une macro à un bouton sur l'une des barres d'outils.

Ceci étant, cela peut provoquer des arrêts intempestifs et des gels de OOo. Par deux fois, sur mon système, cela a fonctionné, puis a ensuite provoqué un gel complet du système. La réparation a nécessité de la chirurgie dans les fichiers de configuration

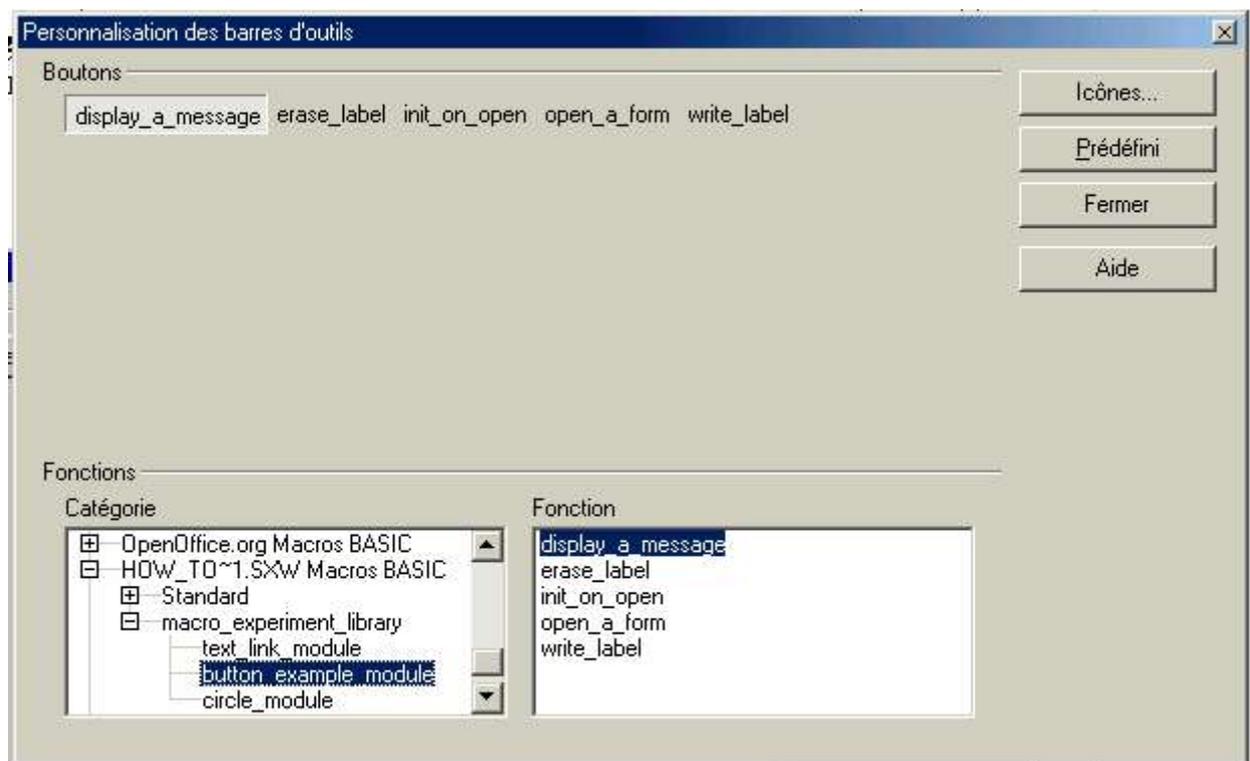
[]/user/config/soffice.cfg/writertoolbar.xml ou

[]/user/config/soffice.cfg/calctoolbar.xml pour découvrir et enlever le composant défectueux, plus quatre redémarrages de Windows-98.

Voici comment cela est censé fonctionner (et le plus souvent, cela fonctionne).

Sélectionnez la barre d'outils à laquelle le bouton doit être ajouté, cliquez avec le bouton droit de la souris sur cette barre, et choisissez **Personnaliser**. Ceci ouvrira la boîte de dialogue de **Personnalisation des Barres d'Outils**.

Dans la fenêtre **Catégorie**, faites défiler jusqu'à la fin de la liste et trouvez les document, bibliothèque, et module contenant la macro que vous voulez lier au bouton. Lorsque vous sélectionnez le module, les noms des macros y étant associés apparaîtront dans la fenêtre **Fonction** et en tant que boutons texte près du haut. Dans la figure, nous avons sélectionné la macro **display_a_message** et le bouton texte du même nom.



Maintenant, il nous faut une petite icône pour remplacer le bouton texte. Cliquez sur le bouton **Boutons...** afin d'afficher une liste de plusieurs centaines de petites icônes qui peuvent servir de bouton de barres d'outils. Ces icônes sont des fichiers bitmap **.bmp**

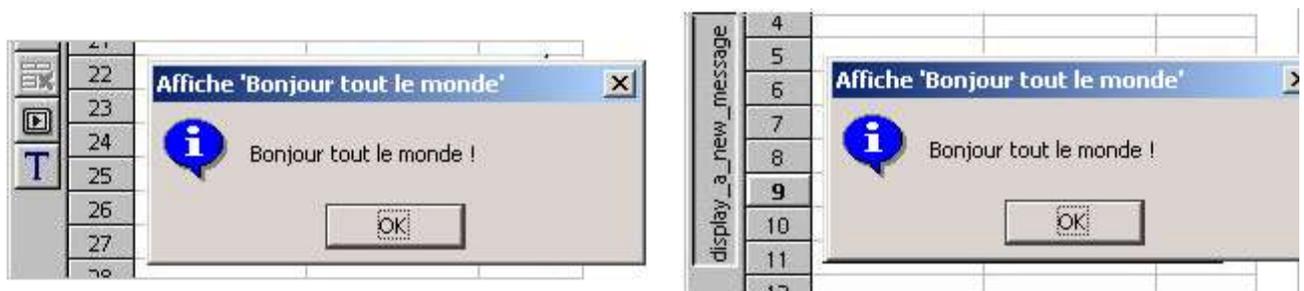
et se trouvent actuellement dans le répertoire [\[install\]/share/config/symbol/](#). Ils portent des noms comme [sc05366.bmp](#), qui seront affichés sous forme de bulles d'aide si vous laissez la souris quelques secondes sur l'icône. Vous pouvez faire vos propres icônes et les rajouter à ce répertoire. Après un redémarrage de OOO, ils apparaîtront dans la liste d'icônes.

Lorsque vous aurez trouver un icône convenable, cliquez dessus, puis sur **OK**. L'icône remplacera le bouton gris de texte dans la boîte de dialogue Personnalisation. Une bulle d'aide vous indiquera la fonction associée au bouton.

Maintenant tirez l'icône depuis la boîte de dialogue Personnaliser et déposez la sur la barre d'outils. Si vous devez la déplacer, utilisez Alt+tirer pour la faire rentrer dans un nouvel endroit sur la barre d'outils ou pour l'enlever, en la laissant tomber en dehors de la barre. Fermez la boîte de dialogue de Personnalisation.

En cliquant maintenant sur ce bouton, vous activerez la macro qui y est associée. Ici, nous avons une macro intitulée [display_a_message](#) que nous avons appelée en cliquant sur le bouton dans la barre d'outils.

Un exemple d'un bouton à icône (**T**) ainsi qu'un bouton texte sont donnés ci-après.



Comme c'est le cas avec les combinaisons de touches, un bouton de barre d'outils reste sur la barre tant qu'il n'est pas enlevé, peu importe les documents ouverts. Si la macro est attribuée à un document en particulier, celle-ci ne sera pas lancée à moins que le document ne soit ouvert. Si vous avez une macro importante que vous désirez lier à un bouton de barre d'outils, nous vous suggérons de la placer dans un module dans le jeu de bibliothèques [soffice](#) de manière à ce qu'elle soit toujours disponible, ou, dans le cas d'un modèle, disponible pour une classe ou un groupe de documents.

Voilà comment cela est censé fonctionner, et c'est comme ça que cela fonctionnera lorsque tous les problèmes ont été corrigés.

Vous pouvez en apprendre plus sur les icônes dans l'aide en ligne. Dans le menu principal, choisissez

Aide > Sommaire > Index et tapez le mot clé **icône** . Consultez les articles **icône; insertion, intégration et adaptation**.

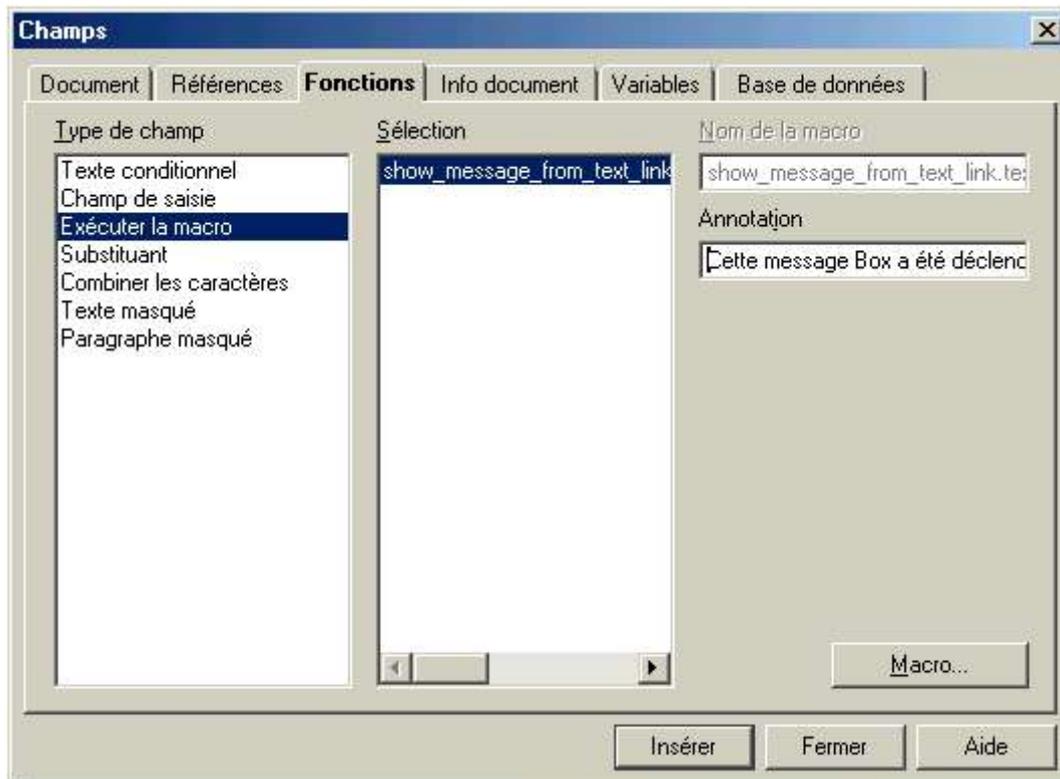
Tapez également dans le champs Recherche le mot clé **barre d'outils** et lisez les divers articles y afférents.

Appeler une Macro avec un Lien Texte

Vous pouvez lier une macro à une sélection de texte, de manière à ce que le texte agit comme un lien pour faire exécuter la macro.

This text acts as a link. Click to launch the macro.

Pour créer cet exemple, choisissez **Insertion > Champ > Autres...** dans le menu principal (ou Ctrl-F2) afin d'ouvrir la boîte de dialogue des **Champs** :



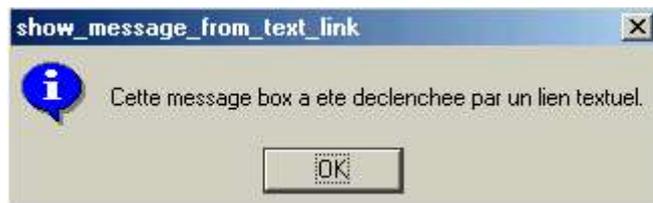
Cliquez sur l'onglet **Fonctions** et dans la fenêtre Type sélectionnez **Exécuter la macro**.

Cliquez sur le bouton **Macro...**, naviguez dans l'arborescence jusqu'à la macro que vous voulez utiliser, et cliquez sur le bouton **Sélectionner...** afin de lier la macro champ de texte.

De retour dans la boîte de dialogue **Champs** , dans le champ **Annotation** , saisissez le texte qui formera le lien.

Cliquez sur **Insérer**, puis **Fermer**.

Désormais, lorsque vous cliquez sur le lien, votre macro sera exécutée :



Appeler une Fonction Macro depuis une Feuille de Calcul

Une macro peut être appelée en tant que formule dans une feuille de calcul. La macro doit être une **Fonction**, et doit renvoyer une valeur numérique ou un chaîne ou une date ou une matrice, c'est-à-dire une valeur que la feuille de calcul saura interpréter.

```
Function le_carré ( x As Double ) As Double
    le_carré = x * x
End Function
```

Les cellules B2:B6 contiennent les valeurs 1..5 et les cellules C2:C6 contiennent la formule **=LE_CARRE(B2)** etc.

	A	B	C	D
1		1	1	
2		2	4	
3		3	9	
4		4	16	
5		5	25	
6				
7				

Ceci est bien entendu une formule simple que l'on aurait pu écrire plus simplement avec **=B2*B2**. Mais la méthode vous permet d'écrire des calculs beaucoup plus complexes qui ne sont pas directement supportés par les fonctions internes de Calc.

Appeler une Macro dans une Autre Bibliothèque

Dans le cas d'une feuille de calcul, la fonction macro devrait certainement être placée dans le document de calcul lui-même, puisqu'elle doit toujours être disponible. Toutefois, vous voudriez peut-être envoyer le document à quelqu'un d'autre. La meilleure solution à ce problème est de placer la fonction macro dans un module quelconque de la **Bibliothèque Standard** du document.

Une feuille de calcul ne semble pas pouvoir reconnaître l'existence d'une fonction résidant dans une autre bibliothèque. Toutefois, une fonction se trouvant dans la bibliothèque standard peut appeler une fonction dans une autre bibliothèque si elle charge au préalable l'autre bibliothèque. Ceci est vrai pour toute macro qui appelle une sous-routine ou une fonction dans une autre bibliothèque, et non seulement des formules de feuilles de calcul.

```
Function   fonc_dans_bib_standard ( X As Double ) As Double
    BasicLibraries.LoadLibrary("nom_de_autre_bibliothèque")
    fonc_dans_bib_standard =   fonc_dans_autre_bib(X)
End Function
```

```
' Pour une bibliothèque dans le conteneur partagé soffice, tel que
'Tools' :
' GlobalScope.BasicLibraries.LoadLibrary("Tools")
```

Toutefois, la bibliothèque n'a besoin d'être chargée **qu'une seule fois**, et non pas à chaque fois que la fonction est appelée. Vous pourriez essayer la fonction suivante. Appelez-la depuis une formule dans une cellule près du début de la feuille de calcul. (L'événement **Document-Open** est activé trop tard pour charger la bibliothèque avant que les cellules ne soient évaluées, et du coup des erreurs se produisent.)

```
Global   library_loaded As Boolean

Function   init( ) As String
```

```

if ( not library_loaded ) then
  BasicLibraries.LoadLibrary("calc_test_library")
  library_loaded = true
end if
init = ""
End Function

```

Si vous utilisez une formule telle que **=some_func(F15)** dans une feuille de calcul, et le résultat qui s'affiche est **#NOM?** ou **#VALEUR?**, alors la fonction n'est pas reconnue. Par exemple :

	A	B	C	D	E	F	G	H	I	J	K	L	M
2													
3		1	in this doc Standard library doc_std_module1										
4													
5			=DOC_STD1_FUNC1(C5)			3.33	36926.04		calls doc_lib1_func1	in calc_test_library			
6			=DOC_STD1_FUNC2(C6)			4.44	406377.68		calls doc_lib1_func2	in calc_test_library			
7													
8		2	in this doc Standard library doc_std_module2										
9													
10			=DOC_STD2_FUNC1(C10)			3.33	36926.04		calls doc_lib1_func1	in calc_test_library			
11			=DOC_STD2_FUNC2(C11)			4.44	1947.02						
12													
13		3	in this doc calc_test_library calc_formula_module										
14													
15			=DOC_LIB1_FUNC1(C15)			3.33	#NAME?		called by doc_std1_func1				
16			=DOC_LIB1_FUNC2(C16)			4.44	#NAME?		called by doc_std1_func2				
17													
18													
19		4	in soffice Standard library soffice_std_module1										
20													
21			=SOFF_STD1_FUNC1(C21)			3.33	100.38		calls soff_lib1_func1	in calc_extra_library			
22			=SOFF_STD1_FUNC2(C22)			4.44	86275.06		calls soff_lib1_func2	in calc_extra_library			
23													
24		5	in soffice Standard library soffice_std_module2										
25													
26			=SOFF_STD2_FUNC1(C26)			3.33	36.93						
27			=SOFF_STD2_FUNC2(C27)			4.44	61.900704						
28													
29		6	in soffice calc_extra_library calc_extra_func_module										
30													
31			=SOFF_LIB1_FUNC1(C31)			3.33	#VALUE!		called by soff_std1_func1				
32			=SOFF_LIB1_FUNC2(C32)			4.44	#VALUE!		called by soff_std1_func2				
33													

Les formules de fonctions macro dans la colonne G des exemples #3 et #6 ne sont pas reconnues en les appelant directement depuis la feuille de calcul, mais les formules dans les exemples #1, #2, et #4 peuvent les appeler de manière indirecte.

Appeler une Macro dans une Validation de Données

Vous pouvez lier une macro à une seule cellule ou à une plage de cellules, de manière à ce que la macro soit appelée si des données non valables sont saisies dans les cellules.

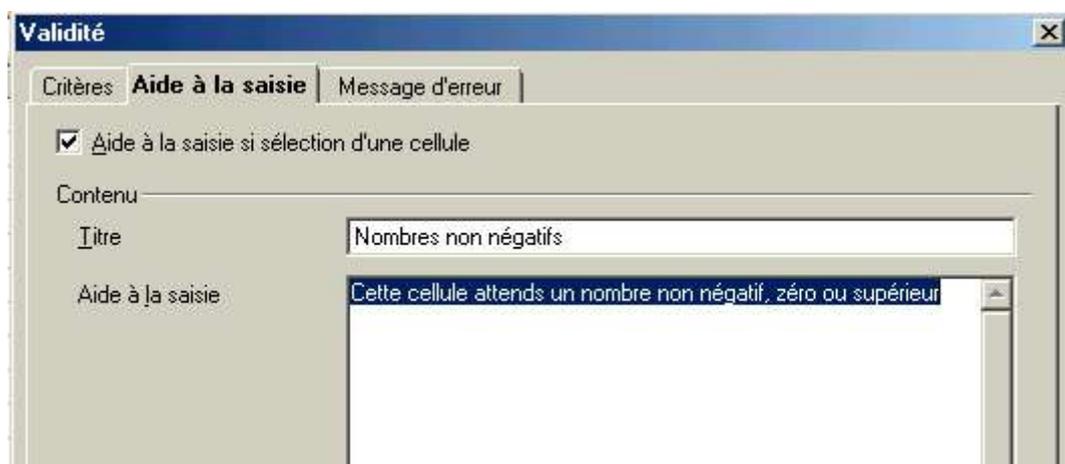
Sélectionnez la cellule ou la plage de cellules et dans le menu choisissez **Données > Validité...** afin d'ouvrir la boîte de dialogue Validité.



On va prendre comme exemple d'exclure des nombres négatifs dans les cellules validées.

Dans l'onglet **Critères**, sous **Autoriser** sélectionnez **Nombre Décimal** et cochez l'option **Autoriser cellules vides**. Sous **Données** sélectionnez **supérieur ou égal à** et tapez **0** pour **Valeur**.

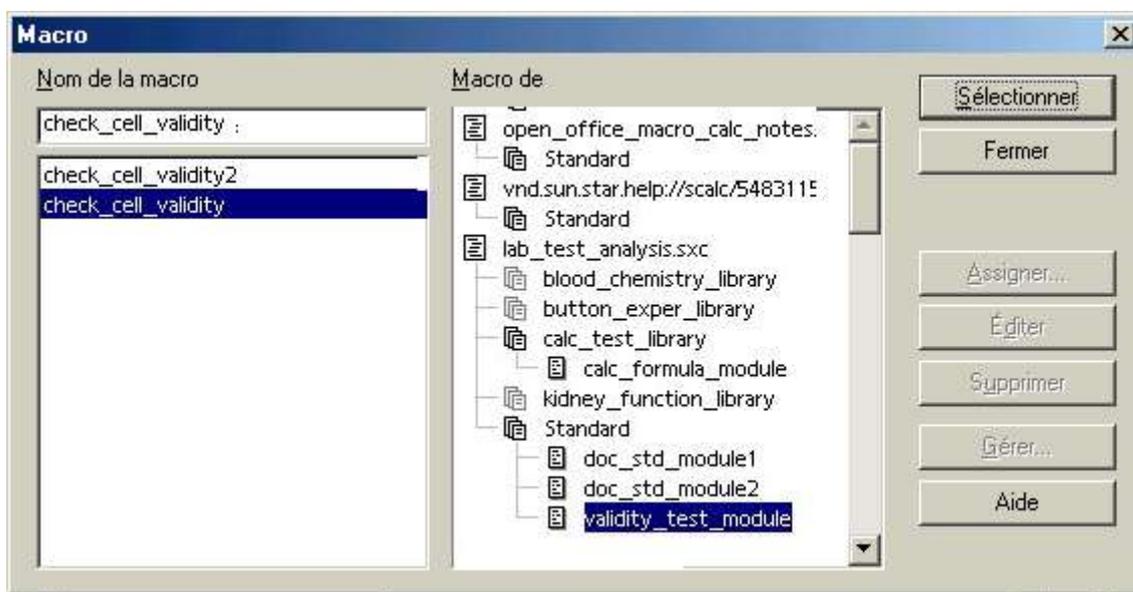
Dans l'onglet **Aide à la Saisie**, cochez la case **Aide à la saisie si sélection d'une cellule** et tapez un titre et une invite pour l'utilisateur (**Aide à la saisie**).



Dans l'onglet **Message d'Erreur**, mettez la valeur de **Action** à **Macro**.



Cliquez sur le bouton **Parcourir...** afin de choisir la macro à lier à la vérification de validité.

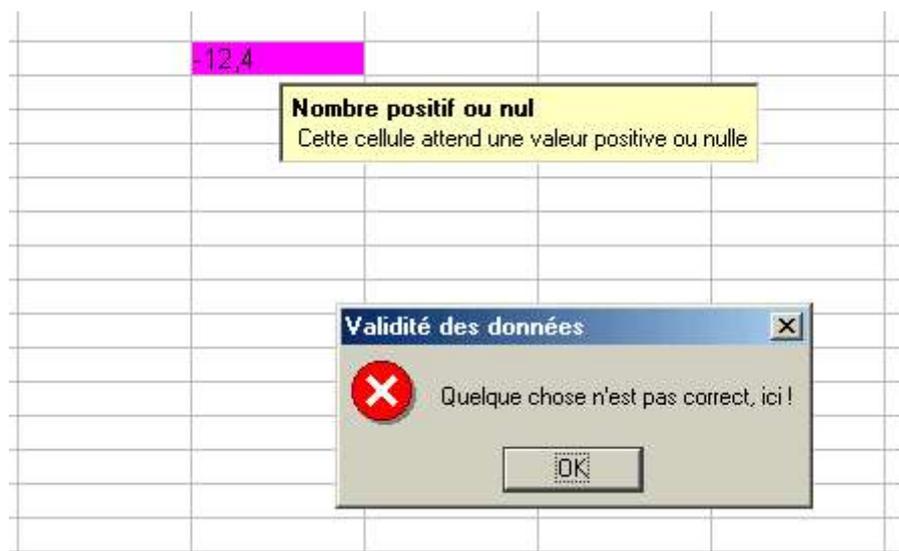


Ici, nous avons sélectionné la macro **check_cell_validity** dans le module **validity_test_module** de notre feuille de calcul.

Cliquer sur le bouton **Sélectionner...** pour lier la macro.

Désormais, lorsque la cellule est sélectionnée, l'utilisateur verra une invite s'afficher (rectangle jaune) expliquant quel type de données il faut saisir.

Et si jamais l'utilisateur saisit une valeur non valable (par exemple un nombre négatif), la macro sera exécutée :



Cette macro n'est qu'un message, mais on pourrait l'écrire autrement pour faire des vérifications plus complexes sur les données saisies. La macro pourrait également être appelée lorsqu'un **changement quelconque** intervient dans la valeur de la cellule.

Vous voudriez peut être empêcher certaines valeurs d'être saisies, selon ce qu'il y a comme données dans les autres cellules.

Un tel exemple serait beaucoup plus qu'une simple validation des données pourrait gérer et nécessiterait une macro.

Création d'une Boîte de Dialogue

Vous pouvez créer une boîte de dialogue dans l'EDI BASIC et l'appeler avec une macro. Cliquez sur le bouton pour faire afficher l'exemple d'une boîte de dialogue.



Si vous voulez reproduire cette boîte de dialogue, ouvrez un nouveau document Writer.

Choisissez **Outils > Macro** pour ouvrir la boîte de dialogue de Macros et cliquez sur le bouton Gérer...

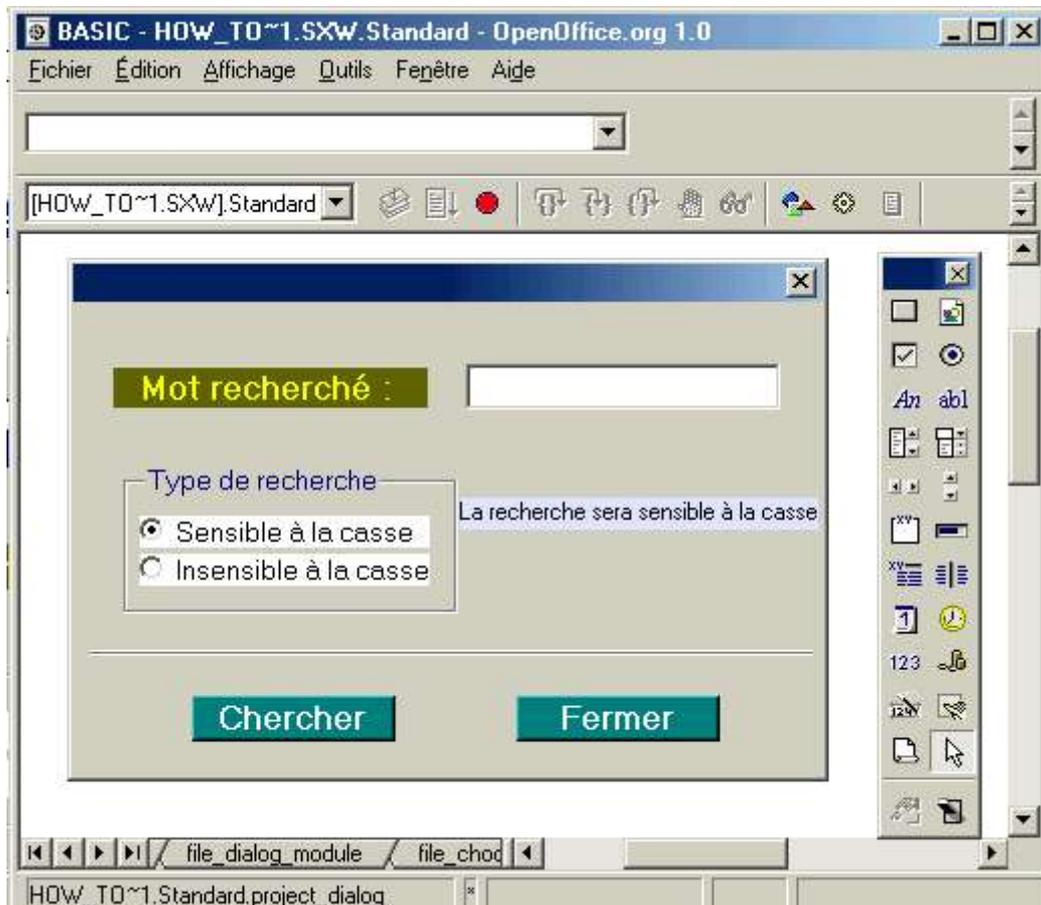
Sélectionnez la bibliothèque **Standard** de votre document.

Cliquez sur **Nouveau Dialogue...** et entrez un nom pour votre nouvelle boîte de dialogue

(la nôtre s'appelle "**project_dialog**").

Une boîte de dialogue vierge apparaît dans l'EDI BASIC.

Dans la deuxième barre d'outils de l'EDI, cliquez sur le bouton **Controls** pour ouvrir la barre d'outils flottante des contrôles de dialogue :



Ici, nous avons utilisé la barre d'outils flottante pour ajouter à la boîte de dialogue vierge des étiquettes, un champ texte, des boutons radio, une ligne horizontale, et deux boutons.

Maintenant, il va falloir des macros pour faire exécuter la boîte de dialogue et réagir en fonction des actions entreprises par les utilisateurs, telles que, par exemple, cliquer sur un bouton ou taper du texte dans un champ de texte dans la boîte de dialogue.

```
Dim oDialogDesc           As Object
Dim oProjectDialog       As Object
Dim oProjectDialogModel As Object
```

```
Sub Open_Project_Dialog
    Dim iDialogResult    As Integer

    ' Obtenir la description de la boîte de dialogue depuis la bibliothèque de
    dialogue
    DialogLibraries.LoadLibrary("Standard")
    oDialogDesc          = DialogLibraries.Standard.project_dialog

    ' créer la boîte de dialogue
    oProjectDialog       = CreateUnoDialog( oDialogDesc )
    oProjectDialogModel = oProjectDialog.Model
    oProjectDialogModel.BackgroundColor = rgb(252, 255, 255)

    ' lancer la boîte de dialogue
    iDialogResult        = oProjectDialog.Execute()

    oProjectDialog.Dispose()
End Sub
```

Ce code simplifié créera et ouvrira la boîte de dialogue. Vous pouvez consulter le code complet en ouvrant la boîte de dialogue des Macros et en étudiant les macros dans ce document.

Ensuite, nous présentons une boîte de dialogue de **parcours de fichier**. Celle-ci vous permet de sélectionner un fichier (mais elle ne tente pas de l'ouvrir).



Open a File

Appeler l'Interface de Programmation d'Applications depuis d'autres Langages

Le Guide du Développeur cité précédemment provenant du projet API fournit de la documentation en détails et des exemples d'écriture de code pour accéder à l'API de OOo API depuis Java, C++, Visual BASIC, et d'autres langages de programmation.

Ceci va toutefois au-delà de ce How-to.

D'autres Exemples

Voici quelques exemples de macros qui ont fait leur apparition sur les différentes listes de discussion. (avec toutes mes excuses aux auteurs dont les noms me sont inconnus.)

Exemple 1.

Voici une macro pour changer la longueur par défaut de la liste des fichiers récemment ouverts, et qui se trouve en dernier du menu Fichier dans le menu principal :

```
REM ***** BASIC *****
Option Explicit

Sub Main
    ChangePickListSize( 10 )
End Sub

Sub ChangePickListSize( nSize as Integer )
    ' accéder au fournisseur de la configuration globale
    Dim aConfigProvider As Object
    aConfigProvider = createUnoService( _
        "com.sun.star.configuration.ConfigurationProvider" )

    ' créer un objet d'accès pour le noeud historique
    Dim aHistorySettings As Object
    Dim aParams(0) As new com.sun.star.beans.PropertyValue
    aParams(0).Name = "nodepath"
    aParams(0).Value = "/org.openoffice.Office.Common/History"
    aHistorySettings = aConfigProvider.CreateInstanceWithArguments( _
        "com.sun.star.configuration.ConfigurationUpdateAccess", _
        aParams() )

    ' fixer la taille de la liste
    aHistorySettings.replaceByName( "PickListSize", nSize )

    ' valider les modifications effectuées
    aHistorySettings.commitChanges
End Sub
```

Bientôt nous disposerons d'une interface utilisateur qui permettra de modifier cette valeur.

Exemple 2.

' Une macro permettant à une touche clavier de modifier les styles

```
Sub ChangeStyle

    Dim oDocument, oDesktop As Object
    Dim oText As Object
    Dim oVCursor, oCursor As Object

    oDesktop = createUnoService("com.sun.star.frame.Desktop")
    oDocument = oDesktop.getCurrentComponent()
    oText = oDocument.Text

    oVCursor = oDocument.currentcontroller.getViewCursor()
    oCursor = oText.createTextCursorByRange(oVCursor.getstart())

    oCursor.paraStyleName = "Default"

End Sub
```

Exemple 3.

' Une macro complète destinée à mettre la date courante dans une cellule sélectionnée

' (incluant le format de nombres, sans la vérification d'erreurs)

' Change TODAY to NOW and DATE to DATETIME

' afin d'afficher la date et l'heure

```
Sub InsertDate
    oDesktop = createUnoService("com.sun.star.frame.Desktop")
    oController = oDesktop.CurrentFrame.Controller
    oSelection = oController.Selection

    ' fixer la valeur de la date
    oDocument = oController.Model
    oFunction = CreateUnoService("com.sun.star.sheet.FunctionAccess")
    oFunction.NullDate = oDocument.NullDate
    dim aEmpty()
    oSelection.Value = oFunction.callFunction("TODAY", aEmpty())

    ' fixer le format de la date
    oFormats = oDocument.NumberFormats
    dim aLocale as new com.sun.star.lang.Locale
    oSelection.NumberFormat = oFormats.getStandardFormat(_
        com.sun.star.util.NumberFormat.DATE, aLocale)

End Sub
```

Exemple 4.

Une macro pour ajouter une liste d'éléments à une boîte de liste dans un formulaire.

```
Sub FillListboxList

    Dim nIndex As Integer
    Dim nMaxIndex As Integer
    Dim n As Integer
    Dim mNameList( nMaxIndex ) As String

    nMaxIndex = 3

    For n = 0 To nMaxIndex

        mNameList( n ) = "Year " + cStr(this_year)

    Next n

    REM Set name list to listbox
    Dim oListboxLB As Object
    oListboxLB = oForm.GetByName( "Listbox1" )
    oListboxLB.StringItemList = mNameList()

End Sub
```

(Les exemples 5, 6, 7, 8, 9, et 10 proviennent de Paolo Mantovani.

A présent, toutes ces macros sont publiées dans la partie Documentation de notre site.
(Ça vaut le coup de les étudier.)

Exemple 5.

Une macro pour copier une plage de cellules dans une feuille de calcul :

```
' -----  
' Routine permettant de copier une plage de cellules de feuille de calcul vers  
une autre  
' la position est définie par rapport à la cellule la plus en haut à gauche  
'  
  
Sub CopyRangeDemo()  
  
    Dim oDocument As Object  
    Dim Spreadsheet As Object  
  
    Set oDocument = ThisComponent  
    Set Spreadsheet = oDocument.Sheets.getByIndex(0)  
  
    REM adresse source (UNO struct)  
    Dim mRangeAddress_src as New com.sun.star.table.CellRangeAddress  
  
    REM adresse cible (UNO struct)  
    Dim mCellAddress_dest as New com.sun.star.table.CellAddress  
  
    REM 2e feuille, C3:D4  
    With mRangeAddress_src  
        .Sheet=1  
        .StartColumn = 2  
        .StartRow = 2  
        .EndColumn = 3  
        .EndRow = 3  
    End With  
    REM 1ère feuille, C3  
    With mCellAddress_dest  
        .Sheet = 0  
        .Column = 2  
        .Row = 2  
    End With  
  
    Spreadsheet.copyRange(mCellAddress_dest, mRangeAddress_src)  
  
End Sub
```

Exemple 6.

Enregistreur de Macro Basic (version anglaise)

Un petit enregistreur de macros pour Calc, écrit en Basic -
contenant plusieurs fonctions destinées à un usage général

http://ooodocs.org/modules.php?name=Downloads&d_op=getit&lid=94

Exemple 7.

TexteVersColonnes (version anglaise)

S'apparente à la fonction d'Excel(r), utile pour séparer du texte sur plusieurs colonnes,
utilisant des délimitants divers

bien commenté! Un bon exemple d'un travail avec une boîte de dialogue et des
services à l'écoute.

Contient plusieurs fonctions à usage général

http://ooodocs.org/modules.php?name=Downloads&d_op=getit&lid=89

Exemple 8.

Changeur de Casse

(La présentation est en italien, mais les macros sont commentées en in anglais)

Une macro qui permet à un utilisateur de changer la casse Majuscule/Minuscule/Titre
d'un texte sélectionné.

Fonctionne bien dans Writer et Calc.

Contient plusieurs fonctions à usage général

http://ooodocs.org/modules.php?name=Downloads&d_op=getit&lid=87

Exemple 9.

Boîte Combo dans une feuille de calcul

(La présentation est en italien, mais les macros sont commentées en anglais)

Excel (r) fournit un contrôle de Boite Combo qui peut interagir directement avec des cellules.

Afin de faire la même chose avec Calc, il vous faut quelques macros.

http://ooodocs.org/modules.php?name=Downloads&d_op=getit&lid=96

Exemple 10.

InLettere (version italienne)

Une fonction Calc, écrite en Basic, qui convertit des nombres en leur correspondant en mots italiens,

par exemple : 1122 ==> millecentoventidue

http://ooodocs.org/modules.php?name=Downloads&d_op=getit&lid=88

Exemple 11.

Il existe plusieurs exemples de macros dans le conteneur de bibliothèques Soffice et sa liste déroulante. Ça vaut le coup de les regarder de plus près.



Certains de ces exemples figurent en français sur notre site, à la rubrique documentation, n'hésitez pas à les consulter

Si vous disposez de macros fonctionnelles bien documentées et commentées, vous êtes vivement incités à les publier sur la liste dev@fr.openoffice.org ou sur la liste doc@fr.openoffice.org pour que d'autres puissent en profiter et apprendre à s'en servir.

Sommaire

Ce How-to est relativement simple.

Vous avez appris comment organiser vos macros dans des bibliothèques et des modules.

Vous avez créé quelques exemples de boutons, touches clavier, et barres d'outils pour appeler des macros.

Vous avez utilisé des fonctions macro dans des formules de feuille de calcul et pour vérifier la validité des données.

Vous avez créé une boîte de dialogue.

Vous comprenez un peu mieux le fonctionnement et l'élaboration de contrôles visuels dans OOo, tels que des propriétés de boutons.

Toutes suggestions et corrections sont les bienvenues.

Credits

Auteur : Daniel Strome

Remerciements: Aux contributeurs des listes de discussion

Intégré par : N/A

Dernières modifications : 30 Septembre 2002

Contacts : OpenOffice.org Documentation <http://documentation.openoffice.org>

Traduction : Alex Thurgood – Adaptation des images Patrick Guimberteau et Serge Le Louarne – Relecture, mise en page Sophie Gautier

Licence

Appendix

Public Documentation License Notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the "License"); you may only use this Documentation if you comply with the terms of this License. A copy of the License is available at <http://www.openoffice.org/licenses/PDL.html>.

The Original Documentation is How to use basic macros. The Initial Writer of the Original Documentation is Daniel Strome Copyright (C) 2002. All Rights Reserved. (Initial Writer contact(s): dan.strome@mpcug.com .

Contributor(s): Alexander Thurgood (traduction) Patrick Guimberteau (adaptation des images) Serge Le louarne (adaptation des images).
Portions created by Alexander Thurgood, Patrick Guimberteau, Serge Le Louarne are Copyright (C) 2002. All Rights Reserved. (Contributor contact(s): alex.thurgood@free.fr pguimber@club-internet.fr serge.le_louarne@tiscali.fr).

NOTE: The text of this **Appendix** may differ slightly from the text of the notices in the files of the Original Documentation. You should use the text of this **Appendix** rather than the text found in the Original Documentation for Your Modifications.