



Guide du débutant 6.4

Chapitre 13 *Débuter avec les macros*

Utiliser l'Enregistreur de Macro... et au-delà

Droits d'auteur

Ce document est protégé par Copyright © 2020 par l'Équipe de Documentation de LibreOffice. Les contributeurs sont nommés ci-dessous. Vous pouvez le distribuer et/ou le modifier sous les termes de la Licence Publique Générale GNU (<https://www.gnu.org/licenses/gpl.html>), version 3 ou ultérieure, ou de la Licence Creative Commons Attribution (<https://creativecommons.org/licenses/by/4.0/>), version 4.0 ou ultérieure.

Toutes les marques déposées citées dans ce guide appartiennent à leurs légitimes propriétaires.

Contributeurs

Ce chapitre est une adaptation mise à jour de *Getting Started with LibreOffice.org 6.0*.

Ont contribué à cette édition

Jean Hollis Weber

Steve Fanning

Kees Kriek

Ont contribué aux éditions précédentes

Andrew Pitonyak

Peter Schofield

Martin Fox

Ron Faile Jr.

Hazel Russman

Olivier Hallot

Valerii Goncharuk

Andrew Jensen

Jean Hollis Weber

Traduction

De cette édition

Traducteurs Jean-Luc Vandemeulebroucke

Relecteurs Philippe Clément

Des éditions précédentes

Traducteurs Christian Chenal

Jean-Luc Vandemeulebroucke

Relecteurs François Olivier

Pierre-Yves Samyn

Philippe Clément

Retours

Veillez adresser tout commentaire ou suggestion concernant ce document à la liste de diffusion de l'Équipe de Documentation : doc@fr.libreoffice.org



Remarque

tout ce que vous envoyez à la liste de diffusion, y compris votre adresse mail et toute autre information personnelle incluse dans le message, est archivé publiquement et ne peut pas être effacé.

Date de publication et version du logiciel

Publié en octobre 2020. Basé sur LibreOffice 6.4.

Le système d'exploitation utilisé pour les copies d'écran est Windows 10 et le jeu d'icônes est Co-libre.

Utiliser LibreOffice sur un Mac

Sur Mac, certaines touches et certains éléments de menu sont différents de ceux utilisés sous Windows ou Linux. Le tableau ci-dessous donne quelques substitutions courantes pour les instructions de ce chapitre. Pour une liste plus détaillée, voyez l'Aide de l'application.

Windows ou Linux	Équivalent Mac	Effet
Sélection du menu Outils > Options	LibreOffice > Préférences	Accès aux options de configuration
<i>Clic droit</i>	<i>Ctrl+clic</i> ou <i>clic droit</i> selon la configuration de l'ordinateur	Ouvre un menu contextuel
<i>Ctrl (Control)</i>	⌘ (Command)	Utilisé avec d'autres touches
<i>F5</i>	Shift+⌘+F5	Ouvre le Navigateur
<i>F11</i>	⌘+T	Ouvre l'onglet <i>Styles et Formatage</i> du volet latéral

Table des matières

Introduction.....	1
Vos premières macros.....	1
Ajouter une macro.....	1
Enregistrer une macro.....	4
Exécuter une macro.....	5
Afficher et éditer des macros.....	6
Commentaires REM.....	7
Définir des sous-routines avec SUB.....	7
Définir des variables avec DIM.....	7
Explication du code de la macro.....	8
Créer une macro.....	9
Un exemple de macro plus compliquée.....	9
Enregistrement de la macro.....	10
Exécuter rapidement une macro.....	13
Limites de l'enregistrement de macro.....	14
Schéma de transmission des commandes.....	14
Comment l'enregistreur de macro utilise-t-il le schéma de transmission des commandes ?.....	14
Autres options.....	15
Organisation des macros.....	15
Où les macros sont-elles enregistrées ?.....	17
Importer des macros.....	17
Télécharger des macros à importer.....	19
Comment exécuter une macro ?.....	19
Barres d'outils, éléments de menus et raccourcis clavier.....	20
Événements.....	20
Les extensions.....	21
Écrire des macros sans l'enregistreur.....	22
Aperçu des macros BeanShell, JavaScript et Python.....	23
Introduction.....	23
Macros BeanShell.....	24
Macros JavaScript.....	25
Macros Python.....	26
Où trouver plus d'informations.....	27
Matériel inclus.....	28
Ressources en ligne.....	28

Introduction

Une macro est une suite de commandes ou d'appuis sur des touches du clavier qui sont conservées pour une utilisation ultérieure. Un exemple de macro simple serait une macro qui "saisit" votre adresse dans un document ouvert. Vous pouvez utiliser des macros pour automatiser des tâches simples ou complexes. Les macros sont particulièrement utiles pour répéter une suite d'actions exactement de la même façon, plusieurs fois au cours du temps.

La façon la plus simple de créer une macro consiste à enregistrer une série d'actions par l'intermédiaire de l'interface utilisateur de LibreOffice. Celui-ci sauvegarde les macros enregistrées en utilisant le langage de script open source LibreOffice Basic, qui est un dialecte du très connu langage de programmation BASIC. Ces macros peuvent être éditées et améliorées après enregistrement grâce à l'Environnement de Développement Intégré (EDI) de LibreOffice Basic.

Les macros les plus puissantes sont créées dans LibreOffice en écrivant leur code grâce à l'un des quatre langages de script acceptés (LibreOffice Basic, BeanShell, JavaScript et Python). Ce chapitre offre un tour d'horizon des fonctionnalités macro de LibreOffice, essentiellement tourné vers son langage de script macro par défaut, LibreOffice Basic. Quelques exemples sont donnés pour les langages de script BeanShell, JavaScript et Python, mais une description plus complète des fonctionnalités de ces langages sort du cadre de cet article.

Vos premières macros

Ajouter une macro

La première étape dans l'apprentissage de la programmation des macros est de savoir retrouver et utiliser des macros existantes. Dans la suite de cette section, vous devez disposer d'une macro que vous voulez utiliser. Pour cet exemple, utilisez la macro simple du Listing 1, qui permet d'afficher le message « Bonjour ! ». Vous devez créer une bibliothèque et un module pour contenir votre macro. Voyez « **Organisation des macros** » page 15 pour plus d'informations.

Listing 1. Macro simple qui affiche bonjour

```
Sub MacroBonjour
  MsgBox "Bonjour !"
End Sub
```

Suivez les étapes suivantes pour créer une bibliothèque qui contiendra votre macro :

- 1) Choisissez **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1).

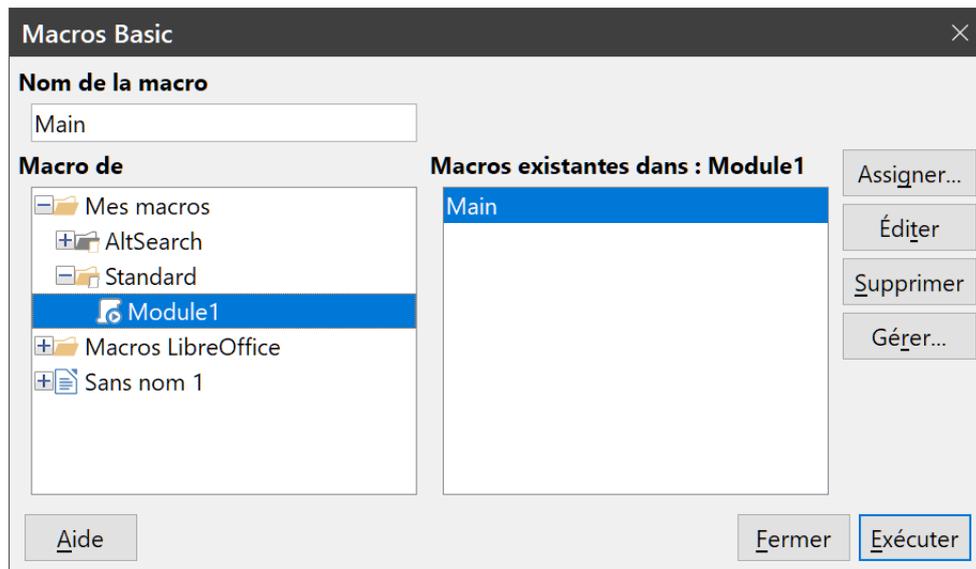


Figure 1 : Boîte de dialogue Macros Basic

- 2) Cliquez sur **Gérer** pour ouvrir la boîte de dialogue *Organiseur de macros Basic* et sélectionnez l'onglet *Bibliothèques* (Figure 2).

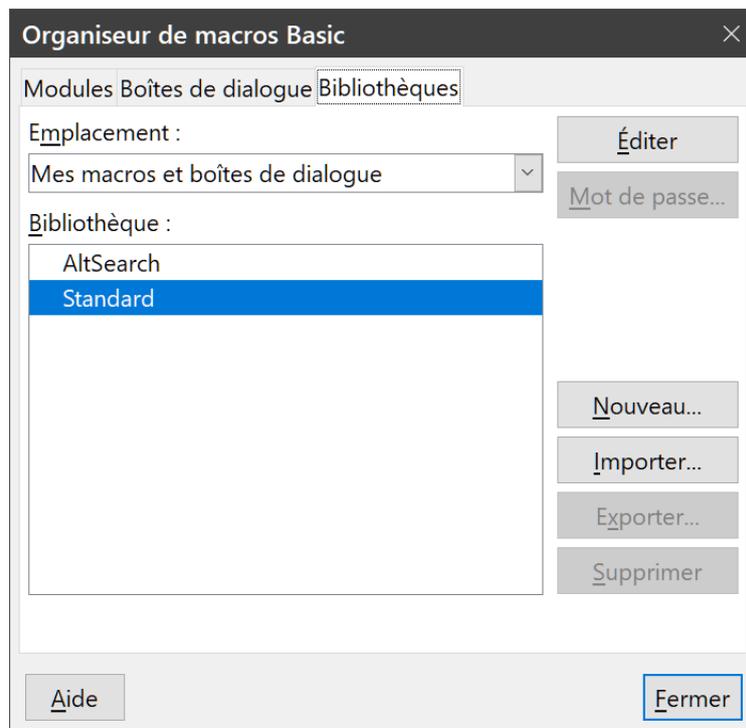


Figure 2 : l'onglet *Bibliothèques* de la boîte de dialogue *Organiseur de macros Basic*

- 3) Si ce n'est déjà fait, choisissez l'emplacement *Mes macros et boîtes de dialogue*, qui est l'emplacement par défaut.
- 4) Cliquez sur **Nouveau** pour ouvrir la boîte de dialogue *Nouvelle bibliothèque*.
- 5) Saisissez le nom de la nouvelle bibliothèque, *BibTest* par exemple, et cliquez sur **OK**.
- 6) Dans la boîte de dialogue *Organiseur de macros Basic*, sélectionnez l'onglet *Modules* (Figure 3).

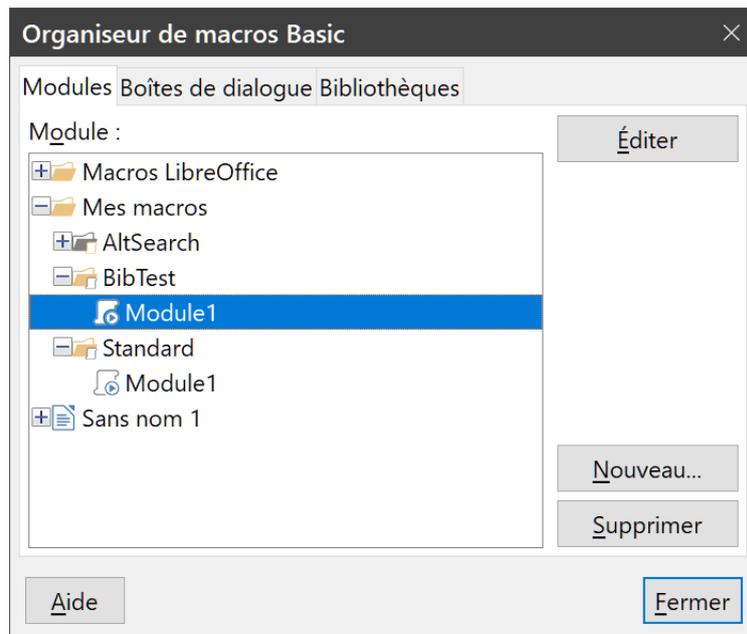


Figure 3 : l'onglet Modules de la boîte de dialogue Organiseur de macros Basic

- 7) Dans la liste des modules, développez *Mes macros* et sélectionnez votre nouvelle bibliothèque. Un module nommé *Module1* existe déjà et peut contenir votre macro. Vous pouvez également cliquer sur **Nouveau** pour créer un nouveau module dans votre bibliothèque.

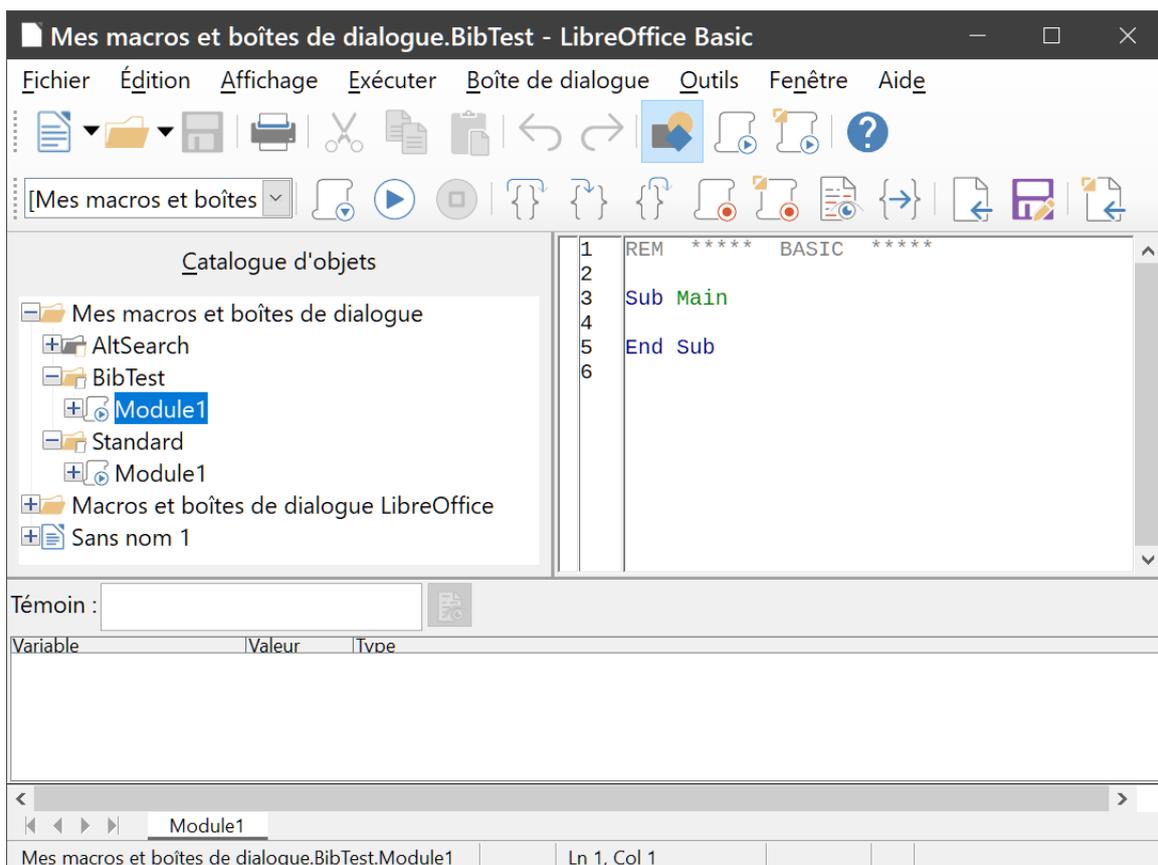


Figure 4 : Environnement de Développement Intégré LibreOffice Basic

- 8) Sélectionnez le module *Module1* et cliquez sur **Éditer** pour ouvrir l'Environnement de Développement Intégré (Figure 4). Cet EDI est un éditeur de texte, disposant de fonctions éten-

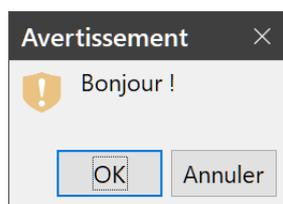
dues, intégré à LibreOffice qui vous permet de créer, de modifier, d'exécuter et de déboguer des macros.

- 9) Lorsqu'un nouveau module est créé, il contient un commentaire et une macro vide nommée `Main`, qui ne fait rien.
- 10) Ajoutez la nouvelle macro avant `Sub Main` ou après `End Sub`. Le Listing 2 montre la nouvelle macro insérée avant `Sub Main`.

Listing 2. Module1 après l'ajout de la macro

```
REM ***** BASIC *****  
  
Sub MacroBonjour  
    Print "Bonjour !"  
End Sub  
  
Sub Main  
  
End Sub
```

- 11) Cliquez sur l'icône **Compiler**  de la barre d'outils *Macro* pour compiler la macro.
- 12) Placez le curseur à l'intérieur de la sous-routine `MacroBonjour` et cliquez sur l'icône **Exécuter**  de la barre d'outils *Macro*, ou appuyez sur la touche *F5*, pour exécuter la macro `MacroBonjour` du module. Une petite boîte de dialogue s'ouvre en affichant le mot `Bonjour`.



- 13) Cliquez sur **OK** pour refermer cette boîte de dialogue.
- 14) Si le curseur ne se trouve pas à l'intérieur d'une sous-routine ou d'une fonction, la boîte de dialogue *Macros LibreOffice Basic* s'ouvre et vous demande de sélectionner la macro à exécuter. L'icône **Sélectionner une macro**  de la barre d'outils *Standard* de l'EDI et la commande **Outils > Macros > Gérer les macros > Basic** ont le même effet.
- 15) Sélectionnez une macro puis cliquez sur le bouton **Exécuter**.

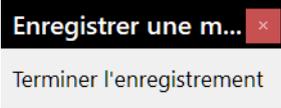
Enregistrer une macro

Si vous devez effectuer des opérations de manière répétée, comme la saisie d'informations fixes par exemple, vous pouvez créer une macro qui mémorise ces opérations. La façon la plus simple de le faire est d'utiliser l'enregistrement de macro.



Remarque

Dans certains cas, si vous devez saisir des informations répétitives dans un document, il peut être plus pratique de créer un autotexte. Voir le chapitre 2, *Travailler avec le texte, les bases*, du *Guide Writer* pour plus d'informations.

- 1) Vérifiez que l'enregistrement de macro est bien actif en ouvrant la page *LibreOffice – Avancé* de la boîte de dialogue *Options* (**Outils > Options** dans la barre de menu) et en cochant l'option *Activer l'enregistreur de macros (peut-être limité)*. Par défaut, cette fonction est désactivée lors de l'installation de LibreOffice sur votre ordinateur.
- 2) Choisissez **Outils > Macros > Enregistrer une macro** dans la barre de menu pour démarrer l'enregistrement de la macro. Une petite boîte de dialogue *Enregistrer une macro* apparaît, indiquant que l'enregistrement de macro est en cours.
- 3) Effectuez la séquence d'opérations que vous désirez. Par exemple, saisissez votre nom.
- 4) Cliquez sur **Terminer l'enregistrement** dans la boîte de dialogue *Enregistrer une macro* pour arrêter l'enregistrement. La boîte de dialogue *Macros Basic* s'ouvre (semblable à celle de la Figure 1 avec deux boutons différents).
- 5) Déroulez le conteneur de bibliothèques *Mes macros* et trouvez la bibliothèque *Standard*. Notez que chaque conteneur de bibliothèques contient une bibliothèque nommée *Standard*.
- 6) Sélectionnez la bibliothèque *Standard* et cliquez sur **Nouveau Module** pour créer le nouveau module qui va contenir la macro. La boîte de dialogue *Nouveau module* s'ouvre.
- 7) Saisissez un nom significatif pour le nouveau module, *Enregistrements* par exemple, et cliquez sur **OK** pour le créer. Le nouveau module figure désormais dans la bibliothèque *Standard* dans la boîte de dialogue *Macros Basic*.
- 8) Dans la zone **Nom de la macro**, saisissez le nom de la macro que vous venez d'enregistrer, *SaisieNom* par exemple.
- 9) Cliquez sur **Enregistrer** pour enregistrer la macro et fermer la boîte de dialogue *Macros Basic*.

Si vous avez suivi toutes les étapes précédentes, la bibliothèque *Standard* contient maintenant un module appelé *Enregistrements* et celui-ci possède une macro nommée *SaisieNom*.



Remarque

Quand LibreOffice crée un nouveau module, il y ajoute automatiquement la macro appelée *Main*.

Exécuter une macro

- 1) Choisissez **Outils > Macros > Exécuter une macro** dans la barre de menu pour ouvrir la boîte de dialogue *Sélecteur de macro* (Figure 5).
- 2) Déroulez l'arborescence des bibliothèques, sélectionnez la macro désirée, *SaisieNom* par exemple, et cliquez sur **Exécuter**.

Vous pouvez également choisir **Outils > Macros > Gérer les macros > Basic** dans la barre de menus principale pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1), sélectionner la macro et cliquer sur **Exécuter**.

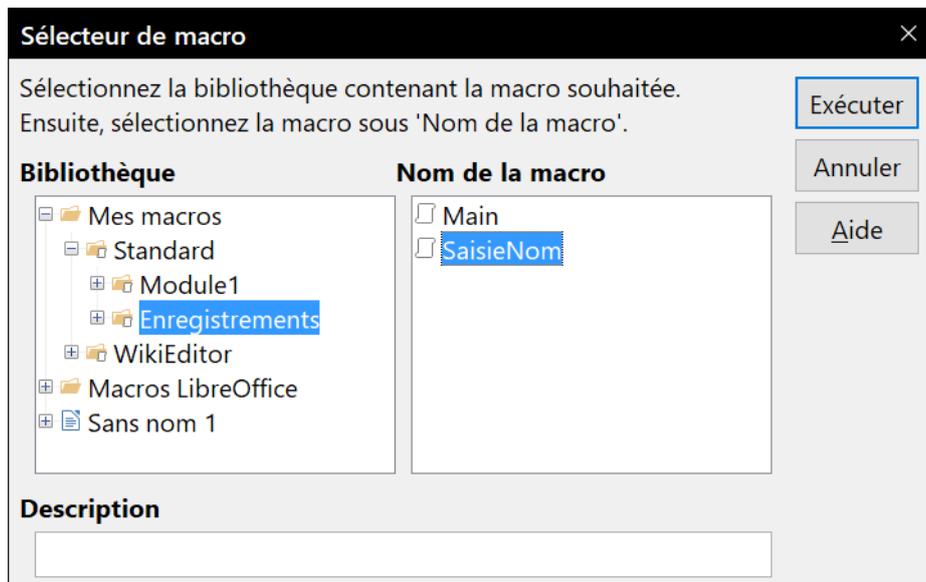


Figure 5 : Boîte de dialogue Sélecteur de macro

Afficher et éditer des macros

Pour afficher ou éditer les macros que vous avez créées :

- 1) Choisissez **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic*.
- 2) Déroulez l'arborescence des bibliothèques, sélectionnez la macro désirée, *SaisieNom* par exemple, et cliquez sur **Éditer** pour ouvrir l'EDI Basic. La macro *SaisieNom* est affichée dans le Listing 3.

Listing 3. Macro enregistrée *SaisieNom*

```

REM ***** BASIC *****
Sub Main

End Sub

sub SaisieNom
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 = "Jean Dupont"

```

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

La macro `SaisieNom` n'est pas aussi compliquée qu'il n'y paraît au premier abord. La connaissance de quelques notions permet de comprendre l'essentiel des macros.

Commentaires REM

Toutes les séquences du code de la macro qui commencent par REM contiennent des commentaires. Tout le texte situé après REM et se trouvant sur la même ligne est ignoré.



Astuce

Vous pouvez employer également le caractère apostrophe (') comme raccourci pour débiter un commentaire.

Les mots-clés de LibreOffice Basic ne sont pas sensibles à la casse : REM, Rem et rem peuvent donc être tous utilisés pour débiter un commentaire. Si vous utilisez des constantes symboliques définies par Interface de Programmation de l'Application (API : Application Programming Interface), il est plus prudent de supposer que les noms sont sensibles à la casse. Les constantes symboliques constituent un sujet plus élaboré qui n'est pas abordé dans ce guide du débutant et n'est pas nécessaire à l'utilisation de l'enregistreur de macros dans LibreOffice.

Définir des sous-routines avec SUB

Chacune des macros est stockée dans une sous-routine. Une sous-routine commence avec le mot-clé SUB, et se termine avec les mots END SUB. Le code commence par la définition de la sous-routine appelée Main, qui est vide et ne fait rien. La sous-routine suivante, `SaisieNom`, contient le code généré pour votre macro.

Les points suivants, qui font partie des fonctionnalités avancées et ne seront pas développés dans ce guide, sont cités pour votre information.

- Vous pouvez écrire une macro de sorte que des valeurs soient transmises à la sous-routine. Ces valeurs sont appelées arguments. Cependant, les macros enregistrées n'admettent pas d'arguments.
- Il existe une autre forme de sous-routine, appelée fonction, qui renvoie une valeur comme résultat de son action. Les fonctions sont définies par le fait qu'elles commencent par le mot-clé FUNCTION. Cependant, l'enregistrement de macro crée toujours des sous-routines qui ne sont pas des fonctions.

Définir des variables avec DIM

Vous pouvez écrire une information sur un bout de papier pour vous en souvenir pour plus tard. Une variable, comme un bout de papier, contient une information qui peut être modifiée et lue. Le mot-clé DIM est à l'origine une abréviation de dimension et était utilisé pour définir les dimensions d'un tableau. L'instruction DIM utilisée dans la macro `SaisieNom` revient à mettre de côté un bout de papier destiné à recueillir un message ou une note.

Dans la macro `SaisieNom`, les variables `document` et `dispatcher` sont définies avec le type `object` (objet). Il existe d'autres types de variables, comme `string` (chaîne de caractères),

integer (nombre entier) ou date (date). Une troisième variable, nommée `args1`, est un tableau de valeurs de propriétés. Une variable de type `array` (tableau) permet à une variable de contenir des valeurs multiples, comme un livre qui contient plusieurs pages. Les valeurs d'un tableau sont habituellement numérotées à partir de zéro. Le nombre entre parenthèses indique le nombre le plus haut utilisable pour accéder à une valeur. Dans cet exemple, il y a uniquement une valeur, et elle porte le numéro zéro.

Explication du code de la macro

Voici une explication du code de la macro `SaisieNom`. Vous ne comprendrez peut-être pas tous les détails, mais l'explication de chaque ligne de code vous donnera une idée de la façon dont la macro fonctionne.

```
sub SaisieNom
```

Définit le début de la macro

```
dim document as object
```

Définit `document` comme objet variable. Les objets sont un type particulier de variable comportant plusieurs champs (parfois appelés propriétés) et des actions (parfois appelées méthodes). Les champs peuvent être vus comme des variables (y compris de type `object`) et les actions comme des sous-routines qui nous permettent d'agir avec l'objet.



Remarque

Le mot `service` est parfois utilisé. Un service est fourni par un type d'objet que l'on distingue afin d'indiquer comment ils sont utilisés.

```
dim dispatcher as object
```

Définit `dispatcher` comme objet variable

```
document = ThisComponent.CurrentController.Frame
```

`ThisComponent` fait référence au document en cours.

`CurrentController` est une propriété qui fait référence au service qui contrôle le document. Par exemple, lorsque vous faites une saisie, c'est `CurrentController` qui mémorise ce que vous tapez. `CurrentController` transmet alors les modifications au cadre du document.

`Frame` est une propriété du contrôleur qui renvoie le cadre principal du document. Par conséquent, `document` est une variable qui fait référence au cadre du document, qui reçoit les commandes transmises.

```
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
```

La plupart des tâches de LibreOffice sont accomplies en transmettant des commandes. LibreOffice comprend un service `DispatchHelper` qui effectue le plus gros du travail pour transmettre les commandes des macros. La méthode `createUnoService` reçoit le nom

d'un service et essaie de créer une instance de ce service. En fin de compte, la variable `dispatcher` fait référence au `DispatchHelper`.

```
dim args1(0) as new com.sun.star.beans.PropertyValue
```

Déclare un tableau de propriétés. Chaque propriété comporte un nom et une valeur. Le tableau créé comporte une seule propriété à l'index zéro.

L'expression `com.sun.star.beans.PropertyValue` est une structure Objets de réseau universels (Universal Network Objects : UNO). Les structures sont un type particulier de variables qui contiennent d'autres variables réunies sur une base logique. Ils peuvent être pratiques pour travailler avec des ensembles d'informations hétérogènes qui doivent être traités dans leur globalité. Une explication des structures UNO et utilisateur sort du cadre de ce livret. Pour plus d'informations sur la création et l'utilisation des structures, voyez l'aide de LibreOffice et les *Antisèches Basic* sur le site où vous avez trouvé ce document.

```
args1(0).Name = "Text"  
args1(0).Value = "Jean Dupont"
```

Donne à la propriété le nom « Text » et la valeur « Jean Dupont », qui est le texte qui est inséré lorsque la macro est exécutée.

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

C'est là que tout se passe. Le `DispatchHelper` transmet au cadre du document (enregistré dans la variable `document`) la commande `.uno:InsertText`. Les deux arguments suivants, le nom du cadre et les flags de recherche, sortent du cadre de ce guide. Le dernier argument est le tableau de propriétés à utiliser lors de la commande `InsertText`.

```
end sub
```

Dernière ligne de code qui termine la sous-routine.

Créer une macro

Lors de la création, il est important de se poser deux questions avant de commencer l'enregistrement :

- 1) Est-ce que cette tâche consiste en un ensemble simple de commandes ?
- 2) Est-ce que les différentes étapes peuvent être agencées de manière à ce que chaque commande laisse le curseur disponible pour la commande suivante ou la saisie de texte ou de données dans le document ?

Un exemple de macro plus compliquée

Vous pouvez avoir besoin de copier des lignes et des colonnes de données à partir d'un site web et de les formater afin d'obtenir un tableau dans un document texte.

- 1) Copiez les données depuis le site web dans le presse-papiers.
- 2) Pour éviter les formats et les polices étranges, copiez le texte du presse-papiers dans un document Writer en tant que texte non formaté.

- 3) Mettez le texte en forme en séparant les colonnes par des tabulations pour qu'il puisse être converti en un tableau grâce à la commande **Tableau > Convertir > Texte en tableau** de la barre de menu.

DONTKNOW	The font weight is not specified/known
THIN	specifies a 50 % font weight.
ULTRALIGHT	specifies a 60 % font weight.
LIGHT	specifies a 75 % font weight.
SEMILIGHT	specifies a 90 % font weight.
NORMAL	specifies a normal font weight.
SEMIBOLD	specifies a 110 % font weight.
BOLD	specifies a 150 % font weight.
ULTRABOLD	specifies a 175 % font weight.
BLACK	specifies a 200 % font weight.

Figure 6 : Texte copié

Enregistrement de la macro

En gardant à l'esprit les deux questions préliminaires ci-dessus, examinez le texte pour voir si une macro peut être enregistrée pour formater le texte. L'exemple de la Figure 6 provient du groupe des constantes de graisse (FontWeight) des polices de caractères dans le site web de l'API. Le nom de la constante est suivi par un espace et une tabulation. La description est suivie de deux espaces.

La première colonne du tableau devrait contenir une valeur numérique, la deuxième le nom et la troisième la description. Cette conversion est facile à réaliser, sauf pour les lignes DONTKNOW et NORMAL qui ne contiennent pas de valeur numérique, mais celles-ci sont comprises entre 0 et 100 et peuvent être saisies manuellement.

Les données peuvent être nettoyées de plusieurs façons. L'exemple ci-dessous utilise les touches du clavier et suppose que le curseur est au début de la ligne commence par le mot THIN.

- 1) Vérifiez que l'enregistrement de macro est bien actif en ouvrant la page *LibreOffice – Avancé* de la boîte de dialogue *Options* (**Outils > Options** dans la barre de menu) et en cochant l'option *Activer l'enregistreur de macros (peut-être limité)*. Par défaut, cette fonction est désactivée lors de l'installation de LibreOffice sur votre ordinateur.
- 2) Positionnez le curseur juste avant le mot THIN.
- 3) Choisissez **Outils > Macros > Enregistrer une macro** pour démarrer l'enregistrement de la macro.
- 4) Appuyez sur *Ctrl+flèche droite* pour déplacer le curseur au début de « specifies ».
- 5) Appuyez sur la touche *Retour arrière* deux fois pour supprimer la tabulation et l'espace.
- 6) Appuyez sur la touche *Tab* pour ajouter une tabulation sans espace après le nom de la constante.

- 7) Appuyez sur la touche *Suppr* pour supprimer le s minuscule, puis appuyez sur *Maj+S* pour ajouter un S majuscule.
- 8) Appuyez sur *Ctrl+flèche droite* deux fois pour déplacer le curseur au début du nombre.
- 9) Appuyez sur *Ctrl+Maj+flèche droite* pour sélectionner le nombre et déplacer le curseur avant le signe %.
- 10) Appuyez sur *Ctrl+C* pour copier le texte sélectionné dans le presse-papiers.
- 11) Appuyez sur la touche *Fin* pour déplacer le curseur à la fin de la ligne.
- 12) Appuyez sur la touche *Retour arrière* deux fois pour supprimer les deux derniers espaces.
- 13) Appuyez sur la touche *Début* pour déplacer le curseur au début de la ligne.
- 14) Appuyez sur *Ctrl+V* pour coller le nombre sélectionné au début de la ligne.
- 15) Coller une valeur ajoute un espace supplémentaire, donc appuyez sur la touche *Retour arrière* pour le supprimer.
- 16) Appuyez sur la touche *Tab* pour ajouter une tabulation entre le nombre et le nom.
- 17) Appuyez sur la touche *Début* pour déplacer le curseur au début de la ligne.
- 18) Appuyez sur la touche *flèche bas* pour déplacer le curseur sur la ligne suivante.
- 19) Arrêtez l'enregistrement et enregistrez la macro (voir « **Enregistrer une macro** » page 4).

Il faut plus de temps pour lire et écrire les différentes étapes que pour enregistrer la macro. Avancez lentement et réfléchissez bien aux étapes au fur et à mesure que vous les effectuez. Avec la pratique, ceci deviendra plus naturel.

Le code généré de la macro du Listing 4 a été modifié, afin de faire apparaître dans les commentaires le numéro de chaque étape ci-dessus.

Listing 4. Copie de la valeur numérique au début de chaque ligne

```

sub FormatLigneTableau
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 (4) Appuyez sur Ctrl+flèche droite pour déplacer le curseur
rem au début de "specifies"
dispatcher.executeDispatch(document, ".uno:GoToNextWord", "", 0, Array())

rem (5) Appuyez sur la touche Retour arrière deux fois pour
rem supprimer la tabulation et l'espace.

```

```

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

rem (6) Appuyez sur la touche Tab pour ajouter une tabulation sans
rem espace après le nom de la constante.
dim args4(0) as new com.sun.star.beans.PropertyValue
args4(0).Name = "Text"
args4(0).Value = CHR$(9)

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

rem (7) Appuyez sur la touche Suppr pour supprimer le s minuscule..
dispatcher.executeDispatch(document, ".uno>Delete", "", 0, Array())

rem (7)... puis appuyez sur Maj+S pour ajouter un S majuscule.
dim args6(0) as new com.sun.star.beans. PropertyValue
args6(0).Name = "Text"
args6(0).Value = "S"

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

rem (8) Appuyez sur Ctrl+flèche droite deux fois pour déplacer le
rem curseur au début du nombre.
dispatcher.executeDispatch(document, ".uno:GoToNextWord", "", 0, Array())
dispatcher.executeDispatch(document, ".uno:GoToNextWord", "", 0, Array())

rem (9) Appuyez sur Ctrl+Maj+flèche droite pour sélectionner le
rem nombre et déplacer le curseur avant le signe %.
dispatcher.executeDispatch(document, ".uno:WordRightSel", "", 0, Array())

rem (10) Appuyez sur Ctrl+C pour copier le texte sélectionné
rem dans le presse-papiers.
dispatcher.executeDispatch(document, ".uno:Copy", "", 0, Array())

rem (11) Appuyez sur la touche Fin pour déplacer le curseur
rem à la fin de la ligne.
dispatcher.executeDispatch(document, ".uno:GoToEndOfLine", "", 0, Array())

rem (12) Appuyez sur la touche Retour arrière deux fois pour
rem supprimer les deux derniers espaces.
dispatcher.executeDispatch(document, ".uno:SwBackspace", "", 0, Array())
dispatcher.executeDispatch(document, ".uno:SwBackspace", "", 0, Array())

rem (13) Appuyez sur la touche Début pour déplacer le curseur
rem au début de la ligne.
dispatcher.executeDispatch(document, ".uno:GoToStartOfLine", "", 0, Array()
)

```

```

rem (14) Appuyez sur Ctrl+V pour coller le nombre sélectionné
rem au début de la ligne.
dispatcher.executeDispatch(document, ".uno:Paste", "", 0, Array())

rem (15)... appuyez sur la touche Retour arrière pour le supprimer.
dispatcher.executeDispatch(document, ".uno:SwBackspace", "", 0, Array())

rem (16) Appuyez sur la touche Tab pour ajouter une tabulation
rem entre le nombre et le nom.
dim args17(0) as new com.sun.star.beans.PropertyValue
args17(0).Name = "Text"
args17(0).Value = CHR$(9)

dispatcher.executeDispatch(document, ".uno:InsertText", "", 0, args17())

rem (17) Appuyez sur la touche Début pour déplacer le curseur
rem au début de la ligne.
dispatcher.executeDispatch(document, ".uno:GoToStartOfLine", "", 0, Array()
)

rem (18) Appuyez sur la touche flèche bas pour déplacer le curseur
rem sur la ligne suivante.
dim args19(1) as new com.sun.star.beans.PropertyValue
args19(0).Name = "Count"
args19(0).Value = 1
args19(1).Name = "Select"
args19(1).Value = false

dispatcher.executeDispatch(document, ".uno:GoDown", "", 0, args19())

end sub

```

Les mouvements de curseur sont utilisés pour toutes les opérations. Si vous exécutez la macro au début de la ligne DONTKNOW, le mot "weight" est copié au début de la ligne et le mot "The" est changé en "She". Ce n'est pas parfait et vous ne devriez pas exécuter sur les lignes qui ont un format différent. Vous devrez formater ces lignes manuellement.

Exécuter rapidement une macro

Il est fastidieux de répéter l'exécution de la macro en utilisant **Outils > Macros > Exécuter une macro** depuis la barre de menu. Faites-le depuis l'EDI Basic. C'est plus facile et rapide, surtout pour les macros temporaires qui ne seront utilisées que quelques fois puis abandonnées.

- 1) Dans le document Writer, placez le curseur au début de la première ligne à traiter.
- 2) Choisissez **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1).
- 3) Déroulez l'arborescence des bibliothèques, sélectionnez la macro désirée et cliquez sur **Éditer** pour ouvrir l'EDI Basic.

- 4) Sauf si vous l'avez changée, la première macro est la macro vide nommée `Main`. Modifiez-la pour qu'elle contienne les instructions du Listing 5.

Vous pouvez désormais exécuter `FormatLigneTableau` en exécutant la macro `Main` par un clic sur l'icône **Exécuter**  dans la barre d'outils *Macro* ou un appui sur la touche `F5`.

- 5) Sélectionnez **Outils > Personnaliser** dans la barre de menu et ouvrez l'onglet *Clavier* de la boîte de dialogue *Personnaliser*.
- 6) Choisissez un raccourci dans la liste *Raccourcis clavier*, puis *Macros de LibreOffice – Mes macros – Standard – Enregistrements* dans la liste *Catégories* et *Main* dans la liste *Fonction*. Terminez en cliquant sur le bouton **Modifier**.

Vous pouvez maintenant exécuter votre macro par le raccourci clavier choisi, ce qui est très rapide et très facile, en particulier pour les macros temporaires qui seront utilisées quelques fois avant d'être abandonnées. Voir « **Comment exécuter une macro ?** » page 19 pour plus d'informations.

Listing 5 : modification de la macro `Main` pour appeler `FormatLigneTableau`

```
Sub Main
    FormatLigneTableau
End Sub
```

Limites de l'enregistrement de macro

L'enregistrement de macro peut parfois ne pas fonctionner complètement. La compréhension des mécanismes internes de LibreOffice permet d'expliquer quand et pourquoi l'enregistrement de macro ne fonctionne pas. Le premier responsable est le schéma de transmission des commandes et sa relation avec l'enregistreur de macros.

Schéma de transmission des commandes

Le but du schéma de transmission des commandes (dispatch framework) est de fournir un accès uniforme aux composants (documents) pour les commandes, qui correspondent habituellement à des éléments de menus. La commande **Fichier > Enregistrer**, le raccourci clavier `Ctrl+S` ou le clic sur l'icône **Enregistrer** dans la barre d'outils *Standard* sont des commandes qui sont traduites dans la même « commande transmise ».

Le schéma de transmission des commandes est également utilisé pour renvoyer des commandes à l'interface utilisateur. Par exemple, après l'enregistrement d'un document, la liste des fichiers récents est mise à jour.

Une commande transmise peut être identifiée par un texte, par exemple `.uno:InsertObject` ou `.uno:GoToStartOfLine`. Cette commande est envoyée au cadre du document et transmise jusqu'à ce qu'un objet soit trouvé pour la prendre en compte.

Comment l'enregistreur de macro utilise-t-il le schéma de transmission des commandes ?

L'enregistrement de macro mémorise la séquence des commandes transmises telles qu'elles sont générées. Cela marche la plupart du temps. Le problème est que certaines commandes transmises ne sont pas complètes. Par exemple, l'insertion d'un objet génère la ligne de code suivante :

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

Il n'est pas possible de spécifier quel type d'objet est créé ou inséré. Si un objet est inséré depuis un fichier, il n'est pas possible de spécifier quel fichier doit être inséré.

Si, lors de l'enregistrement d'une macro, vous utilisez la commande **Outils > Options** pour ouvrir et modifier des éléments de configuration, la macro générée n'enregistrera aucune modification. En fait, le code généré est mis en commentaire, donc il ne sera jamais exécuté.

```
rem dispatcher.executeDispatch(document, ".uno :OptionsTreeDialog », "",  
0, Array())
```

Si une boîte de dialogue est ouverte, une commande pour ouvrir cette boîte sera probablement générée et tout ce qui est effectué à l'intérieur de la boîte ne sera habituellement pas enregistré. C'est le cas par exemple des boîtes de dialogue d'organisation des macros, d'insertion de caractères spéciaux ou d'autres. D'autres problèmes possibles lors de l'enregistrement de macro affectent notamment l'insertion de formule, le paramétrage des données utilisateur, les filtres de Calc, les actions sur les formulaires de base de données et l'export des documents vers des PDF chiffrés. Vous ne pouvez jamais être sûr du bon fonctionnement avant de l'essayer. Par exemple, les actions dans la boîte de dialogue de recherche sont correctement enregistrées.

Autres options

Si l'enregistrement de macro n'est pas capable de résoudre un problème particulier, la solution est d'écrire directement le code en utilisant les objets de LibreOffice. Cependant, cela représente un niveau d'expertise supplémentaire. Il est recommandé de commencer avec des exemples simples et d'augmenter le périmètre des macros au fur et à mesure de votre apprentissage. La lecture des macros générées est un bon début.

Organisation des macros

Dans LibreOffice, les macros sont regroupées en modules, les modules sont regroupés en bibliothèques, et les bibliothèques sont regroupées dans des conteneurs de bibliothèques. Une bibliothèque est habituellement utilisée comme principal niveau de regroupement pour contenir toute une catégorie de macros ou toute une application. Les modules sont généralement utilisés pour regrouper des macros selon leurs fonctionnalités, comme l'interface avec les utilisateurs ou les calculs. Chaque macro peut être une sous-routine ou une fonction. La Figure 7 montre un exemple de structure hiérarchique de bibliothèques de macros dans LibreOffice.

Choisissez **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1). Tous les conteneurs de bibliothèques disponibles sont affichés dans la zone *Macro de*. Chaque document est un conteneur de bibliothèques, qui peut contenir plusieurs bibliothèques. L'application elle-même comporte deux conteneurs de bibliothèques, l'un appelé *Macros LibreOffice* qui contient des macros livrées avec LibreOffice, et l'autre, appelé *Mes macros*, destiné à contenir les macros personnelles.

Les macros de *Macros LibreOffice* sont stockées avec le code exécutable de l'application et peuvent donc ne pas être modifiables si vous n'êtes pas administrateur de l'ordinateur. Ceci permet de protéger ces macros qui ne devraient pas être modifiées. Il est également fortement déconseillé de stocker vos propres macros dans ce conteneur.

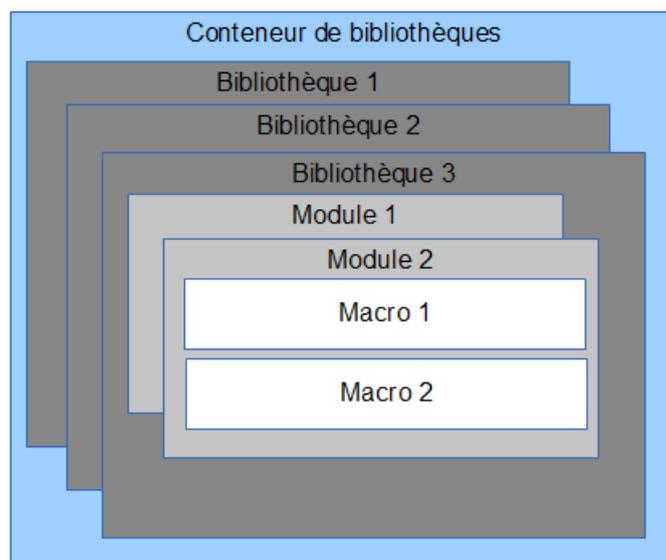


Figure 7 : Hiérarchie des bibliothèques de macros

À moins que vos macros ne soient applicables qu'à un et un seul document, vous les stockerez probablement dans le conteneur *Mes macros*. Ce conteneur est stocké dans votre zone utilisateur ou dans votre dossier par défaut.

Si une macro est contenue dans un document, une macro enregistrée va alors essayer de travailler avec ce document, à travers l'usage de « `ThisComponent` » pour ses actions.

Chaque conteneur de bibliothèques contient une bibliothèque nommée *Standard*. Il est préférable de créer vos propres bibliothèques et de leur donner des noms significatifs, plutôt que d'utiliser la bibliothèque *Standard*. Non seulement les noms significatifs sont plus faciles à utiliser, mais ils peuvent également être importés dans d'autres conteneurs de bibliothèques, alors que ce n'est pas possible avec la bibliothèque *Standard*.



Attention

LibreOffice vous permet d'importer des bibliothèques dans un conteneur de bibliothèques, mais pas d'écraser la bibliothèque nommée *Standard* si elle n'est pas vide. Par conséquent, si vous stockez vos macros dans la bibliothèque *Standard*, vous ne pouvez pas les importer dans un autre conteneur de bibliothèques.

De même qu'il est préférable de donner des noms significatifs à vos bibliothèques, il est prudent de faire de même pour vos modules. Par défaut, LibreOffice utilise les noms `Module1`, `Module2`, et ainsi de suite.

Lorsque vous créez vos macros, vous devez décider où les stocker. Enregistrer une macro dans un document est utile si le document est destiné à être partagé et si vous voulez que la macro suive ce document. Cependant, les macros stockées dans le conteneur de bibliothèques *Mes macros* sont disponibles dans tous les documents.

Les macros ne sont pas disponibles jusqu'à ce que la bibliothèque qui les contient soit chargée. Cependant, la bibliothèque *Standard* est automatiquement chargée. Une bibliothèque s'affiche différemment selon le fait qu'elle soit chargée ou non. Pour charger une bibliothèque et les modules qu'elle contient, double-cliquez sur la bibliothèque.

Où les macros sont-elles enregistrées ?

LibreOffice enregistre les données particulières à un utilisateur dans un sous-dossier du dossier principal de cet utilisateur. Cet emplacement dépend du système d'exploitation. Vous pouvez en avoir un aperçu dans la page *LibreOffice – Chemins* de la boîte de dialogue *Options* (**Outils > Options** dans la barre de menu) et en regardant où les autres données de configuration, les données d'autocorrection par exemple, sont enregistrées. Les macros du conteneur *Mes macros* écrites en Basic sont stockées dans le dossier LibreOffice\4\user\basic. Chaque bibliothèque est stockée dans son propre dossier.

Il n'est pas fondamental de comprendre où les macros sont stockées pour pouvoir les utiliser. Cependant, si vous le savez, vous pouvez les sauvegarder, les partager et les vérifier en cas d'erreur.

Sélectionnez **Outils > Macros > Gérer les boîtes de dialogue** pour ouvrir la boîte de dialogue *Organiseur de macros Basic* (Figure 2, page 2). Vous pouvez aussi sélectionner **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1, page 2) puis cliquer sur le bouton **Gérer**.

Importer des macros

La boîte de dialogue *Organiseur de macros Basic* vous permet d'importer des bibliothèques de macros dans votre document, ainsi que de créer, de supprimer et de renommer des bibliothèques, des modules et des boîtes de dialogue.



Remarque

Vous ne pouvez pas importer la bibliothèque nommée *Standard*.

- 1) Dans l'onglet *Bibliothèques*, choisissez le conteneur de bibliothèques où vous voulez effectuer l'import des macros et cliquez sur le bouton **Importer**.
- 2) Dans la boîte de dialogue *Importer des bibliothèques*, naviguez jusqu'à l'emplacement contenant la bibliothèque à importer. Il y a normalement deux fichiers à importer, *dialog.xlc* et *script.xlc*. Peu importe celui que vous choisirez, ils seront importés tous les deux. Les macros peuvent être stockées dans des bibliothèques à l'intérieur des documents de LibreOffice. Sélectionnez un document plutôt qu'un répertoire sur un disque pour importer les bibliothèques contenues dans un document.



Astuce

Sous Linux, les fichiers propres à LibreOffice sont stockés dans le répertoire home de l'utilisateur, dans un sous-répertoire dont le nom commence par un point (normalement *.config/*). Les répertoires et fichiers dont le nom commence par un point peuvent être cachés et ne pas apparaître dans une boîte de dialogue de sélection normale. Si vous utilisez les boîtes de dialogue de LibreOffice plutôt que celles qui sont propres au système d'exploitation, saisissez directement le nom du répertoire désiré dans le champ *Nom*.

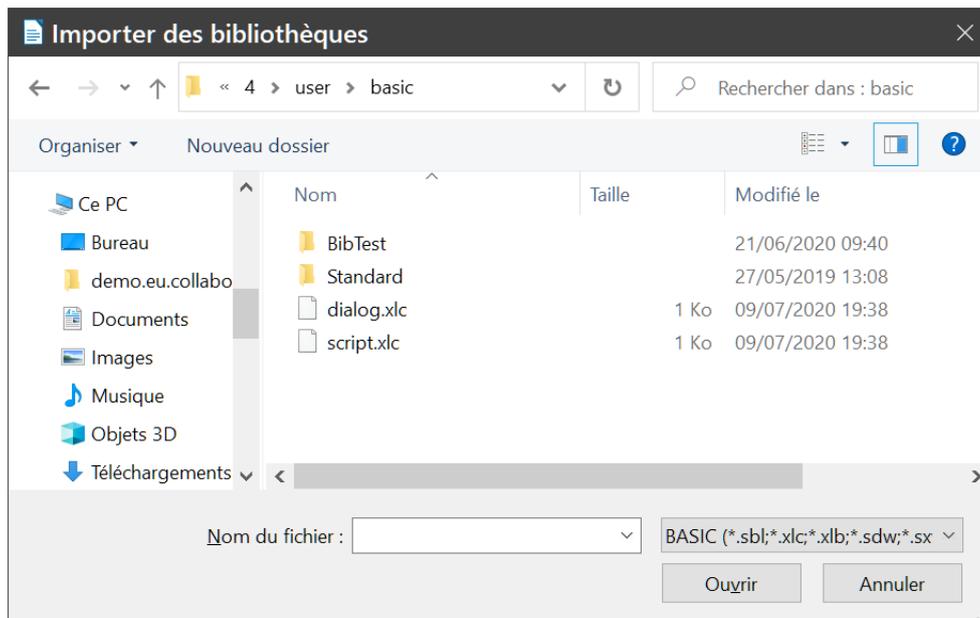


Figure 8 : la boîte de dialogue d'importation des bibliothèques

- 3) Sélectionnez le fichier et cliquez sur le bouton **Ouvrir** pour ouvrir la boîte de dialogue *Importer des bibliothèques* (Figure 9).

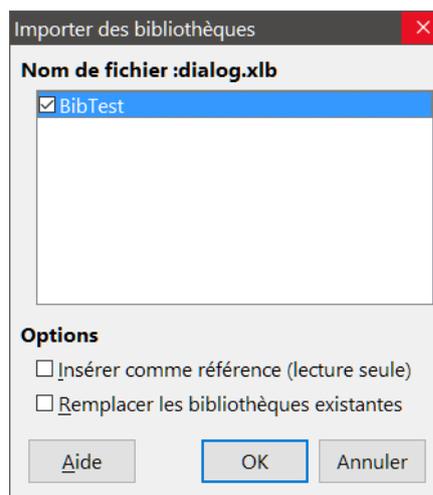


Figure 9 : Choix des options d'import

- 4) Choisissez l'une des options suivantes pour importer les bibliothèques :
- Si aucune option n'est sélectionnée, les bibliothèques sont copiées dans votre conteneur. Cependant, s'il existe déjà une bibliothèque ayant un nom identique dans le conteneur de destination, cette bibliothèque ne sera pas copiée.
 - Cochez *Remplacer les bibliothèques existantes* si vous voulez écraser les bibliothèques existantes dans le conteneur de destination.
 - Cochez *Insérer comme référence (lecture seule)* si vous voulez utiliser la bibliothèque comme référence, mais ne pas l'importer dans le document. Dans ce cas, la bibliothèque reste à son emplacement initial et est en lecture seule.
- 5) Cliquez sur **OK** pour importer les bibliothèques de macros que vous avez sélectionnées.

Télécharger des macros à importer

Des macros peuvent être téléchargées. Certaines sont contenues dans des documents, certaines sont des fichiers normaux à sélectionner et importer, d'autres sont du texte à copier et coller dans l'EDI Basic. Voyez « **Ajouter une macro** », page 1, pour ajouter des macros à votre bibliothèque et « **Afficher et éditer des macros** », page 6, pour les éditer avec l'EDI Basic.

Certaines sont disponibles en téléchargement libre sur internet (voir le Tableau 1). Les sites sont en anglais, sauf mention contraire.

Tableau 1 : exemples de macros

Adresse	Description
http://www.pitonyak.org/oo.php	Documents de référence concernant les macros.
http://www.pitonyak.org/database/	Documents de référence concernant les macros pour base de données.
https://wiki.documentfoundation.org/Macros/fr	Liens vers de la documentation, des outils et des exemples de macro (en français).
http://forum.openoffice.org/en/forum/	Forums avec de nombreux exemples et de l'aide.

Comment exécuter une macro ?

Vous pouvez utiliser **Outils > Macros > Exécuter la macro** pour exécuter toutes les macros, mais ceci peut ne pas être très efficace si vous les utilisez fréquemment. Voir « **Exécuter une macro** » page 5 pour plus d'informations.

Une technique répandue pour les macros fréquemment utilisées est d'attribuer une macro à une icône de barre d'outils, à un élément de menu, à un raccourci clavier ou à un bouton contenu dans le document. Pour choisir une méthode, vous devez vous poser les questions suivantes :

- La macro doit-elle être disponible uniquement dans un document, ou globalement ?
- Est-ce que la macro est spécifique à un type de document, comme les documents Calc par exemple ?
- À quelle fréquence la macro sera-t-elle utilisée ?

Les réponses vont déterminer où ranger la macro et comment la rendre disponible. Par exemple, vous n'incorporez pas une macro rarement utilisée dans une barre d'outils. Pour vous aider à choisir, voir Tableau 2.

Tableau 2. Méthodes pour appeler une macro

Type	LibreOffice	Type de document	Document
Barre d'outils	Non	Oui	Oui
Menu	Non	Oui	Oui
Raccourci	Oui	Oui	Non
Événement	Oui	Non	Oui

Barres d'outils, éléments de menus et raccourcis clavier

Pour ajouter un élément de menu, un raccourci clavier ou une icône de barre d'outils qui appelle une macro, utilisez la boîte de dialogue *Personnaliser*.

Pour ouvrir cette boîte de dialogue, choisissez **Outils > Personnaliser** dans la barre de menu. Elle offre des onglets pour configurer les menus, les raccourcis clavier, les barres d'outils et les événements, dont l'utilisation est présentée au chapitre 14, *Personnaliser LibreOffice*.

Événements

Dans LibreOffice, quand quelque chose se passe, cela signifie qu'un événement s'est produit : par exemple, l'ouverture d'un document, l'appui d'une touche ou le déplacement du pointeur de la souris. LibreOffice permet que des événements puissent déclencher l'appel d'une macro ; la macro est alors appelée gestionnaire d'événement. L'étude complète des gestionnaires d'événement dépasse le cadre de ce chapitre, quelques connaissances peuvent être nécessaires.



Attention

Faites attention quand vous configurez un gestionnaire d'événement. Par exemple, si vous avez configuré un gestionnaire d'événement appelé à chaque fois que le document est modifié. Si l'événement n'est pas correctement géré, vous pouvez être amené à arrêter de force LibreOffice.

- 1) Choisissez **Outils > Personnaliser** dans la barre de menus pour ouvrir la boîte de dialogue *Personnaliser* et sélectionner l'onglet *Événements* (Figure 10). Les événements qui apparaissent ici concernent l'application tout entière ainsi que des documents spécifiques.
- 2) Dans la liste déroulante *Enregistrer dans*, choisissez LibreOffice ou un document précis.
- 3) Un usage courant consiste à affecter une macro à l'événement *Ouvrir le document*, afin d'effectuer un certain nombre de tâches d'initialisation pour un document. Sélectionnez l'événement désiré et cliquez sur le bouton **Macro** pour ouvrir la boîte de dialogue *Sélecteur de macro* (semblable à la Figure 5, page 6, mais avec des boutons différents).
- 4) Sélectionnez la macro désirée et cliquez sur **OK** pour affecter la macro à l'événement. L'onglet *Événements* est alors mis à jour (Figure 10) pour montrer qu'une macro a été affectée à l'événement.

Un certain nombre d'objets dans un document peuvent être paramétrés pour appeler une macro quand un événement se produit. Voir les guides de chaque composant pour plus de détails. La façon la plus ordinaire est d'ajouter un contrôle, comme un bouton par exemple, qui, lorsque vous le cliquez, appelle une macro. Même un double-clic sur une image ouvre une boîte de dialogue avec un onglet *Macros* qui vous permet d'affecter une macro à un événement.

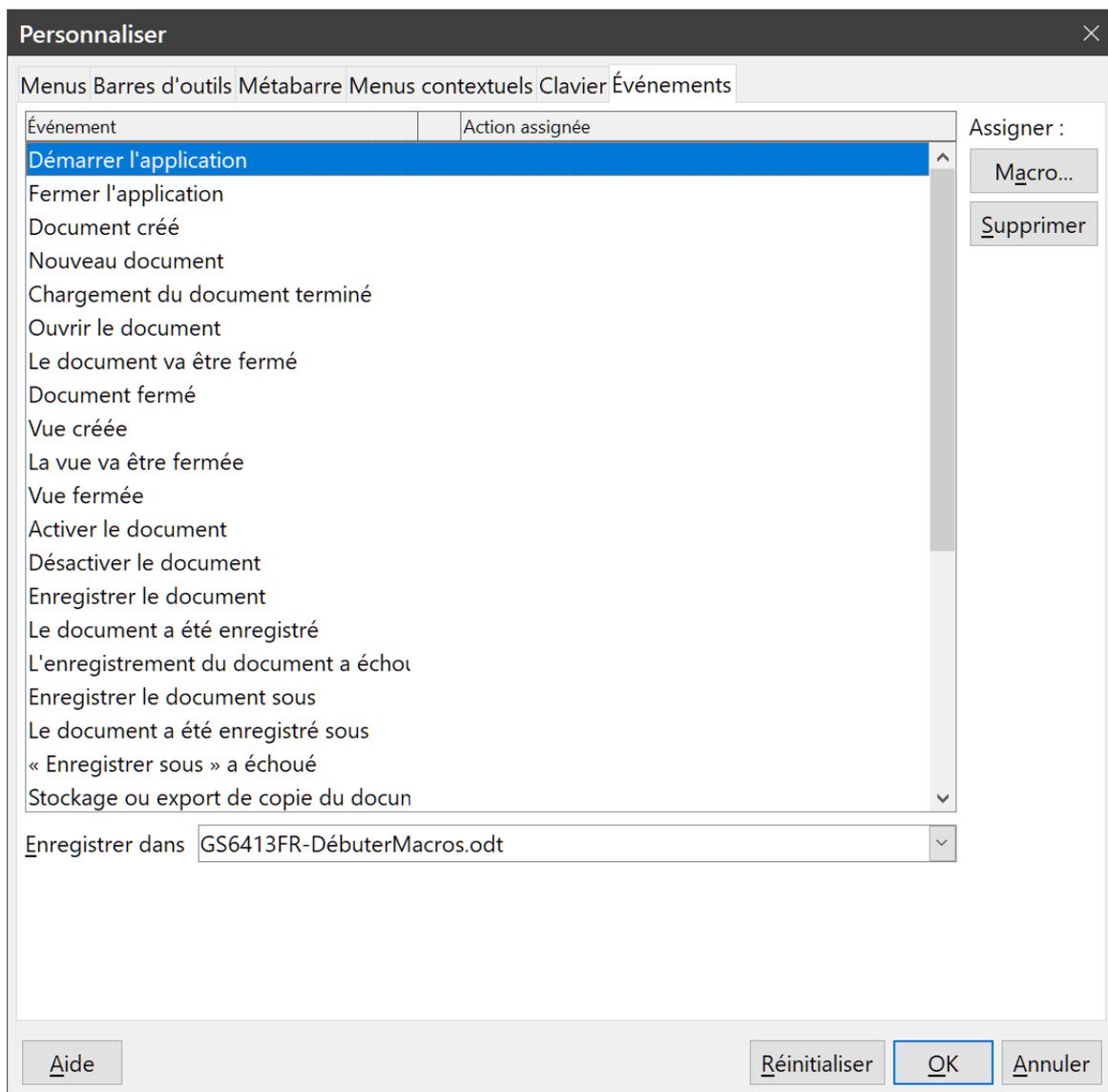


Figure 10 : Onglet Événements de la boîte de dialogue Personnaliser

Les extensions

Une extension est un paquet logiciel qui peut être installé dans LibreOffice pour lui ajouter de nouvelles fonctionnalités. Les extensions peuvent être écrites avec presque n'importe quel langage et peuvent être simples aussi bien que sophistiquées. On peut les regrouper en différents types :

- les modules d'extension de Calc, qui lui apportent de nouvelles fonctionnalités, y compris de nouvelles fonctions qui se comportent comme les fonctions incorporées ;
- de nouveaux composants et de nouvelles fonctionnalités qui s'intègrent normalement à un certain endroit de l'interface utilisateur, comme de nouveaux menus ou de nouvelles barres d'outils ;
- des extensions qui proposent de nouveaux types de diagrammes ; les composants linguistiques, tels que des dictionnaires orthographiques ;
- les modèles de documents et des images.

Bien qu'on puisse les trouver à différents endroits, il existe un dépôt spécifique pour les extensions : <https://extensions.libreoffice.org/>. Vous pouvez aussi trouver de la documentation à <https://libreplanet.org/wiki/Group:OpenOfficeExtensions/List> (en anglais).

Pour plus d'informations sur les extensions et leur installation, voir le chapitre 14, *Personnaliser LibreOffice*.

Écrire des macros sans l'enregistreur

Les exemples précédents ont été créés en utilisant l'enregistreur de macro. Vous pouvez également écrire des macros qui accèdent directement aux objets de LibreOffice si vous vous sentez capables d'écrire du code informatique. En d'autres termes, vous pouvez créer une macro qui manipule directement le document.

La manipulation directe des objets internes de LibreOffice est une fonction avancée qui va au-delà du propos de ce chapitre. Voici toutefois un exemple simple pour illustrer ce fonctionnement.

Listing 6. Ajout du texte "Bonjour" à la fin du document en cours

```
Sub AjoutBonjour
  Dim mDoc
  Dim sTextService$
  Dim mCurs

  REM ThisComponent est le document en cours
  mDoc = ThisComponent

  REM Vérifie que c'est un document texte
  sTextService = "com.sun.star.text.TextDocument"
  If NOT mDoc.supportsService(sTextService) Then
    MsgBox "Cette macro ne fonctionne qu'avec un document texte"
    Exit Sub
  End If

  REM Récupère le curseur en cours
  mCurs = mDoc.currentController.getViewCursor()

  REM Déplace le curseur à la fin du document
  mCurs.gotoEnd(False)

  REM Insère le texte "Bonjour" à la fin du document
  mCurs.Text.insertString(mCurs, "Bonjour", False)
End Sub
```

Aperçu des macros BeanShell, JavaScript et Python

Introduction

Comme de nombreux programmeurs peuvent ne pas être familiers de LibreOffice Basic, Calc permet d'écrire des macros dans trois autres langages qui sont peut-être plus connus. Ce sont BeanShell, JavaScript et Python.

Les macros sont organisées de la même manière pour les quatre langages de script. Le conteneur *LibreOffice Macros* contient toutes les macros fournies à l'installation de LibreOffice. La bibliothèque *Mes Macros* contient vos macros qui sont disponibles dans n'importe lequel de vos documents LibreOffice. Chaque document peut aussi contenir vos macros qui ne sont disponibles dans aucun autre document.

Quand vous utilisez la fonction **Enregistreur de macro**, LibreOffice crée la macro en LibreOffice Basic. Pour utiliser les autres langages de script disponibles, vous devez écrire le code vous-même.

Quand vous sélectionnez **Outils > Macros > Exécuter la macro** dans la barre de menu, LibreOffice ouvre la boîte de dialogue *Sélecteur de macro* qui vous permet de choisir et d'exécuter n'importe quelle macro disponible, quel que soit le langage qui a servi à la coder (Figure 11).

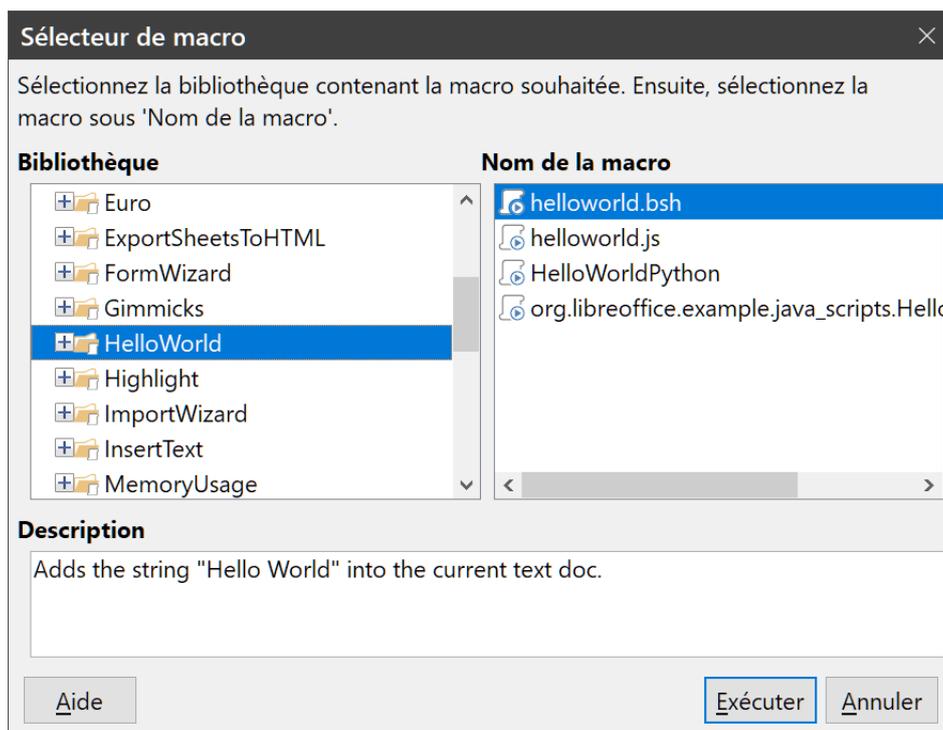


Figure 11 : la boîte de dialogue *Sélecteur de macro*.

Quand vous sélectionnez **Outils > Macros > Éditer la macro** dans la barre de menu, LibreOffice ouvre l'EDI de LibreOffice Basic qui vous permet de choisir et de modifier toute macro LibreOffice Basic disponible, mais pas celles écrites dans d'autres langages.

Le modèle de composants utilisé dans LibreOffice est connu sous le nom Universal Network Objects (UNO). Les macros de LibreOffice, écrites avec n'importe quel langage de script, utilisent à l'exécution une interface de programmation d'application (Application Programming Interface : API) de UNO. L'interface XSCRIPTCONTEXT est transmise aux scripts des macros dans les quatre

langages et offre un moyen d'accéder aux diverses interfaces dont vous pouvez avoir besoin pour effectuer une action sur un document.

Macros BeanShell

BeanShell est un langage de script qui ressemble à Java sorti en 1999.

Quand vous sélectionnez **Outils > Macros > Gérer les macros > BeanShell** dans la barre de menu, Calc ouvre la boîte de dialogue *Macros BeanShell* (Figure 12).

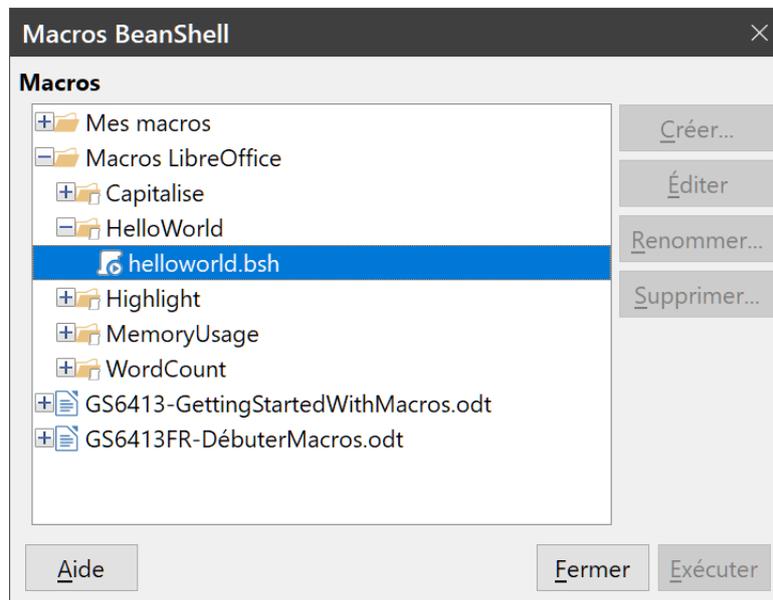


Figure 12 : la boîte de dialogue *Macros BeanShell*.

Cliquez sur le bouton **Éditer** de la boîte de dialogue *Macros BeanShell* pour ouvrir la boîte de dialogue *BeanShell Debug Window* (Figure 13).

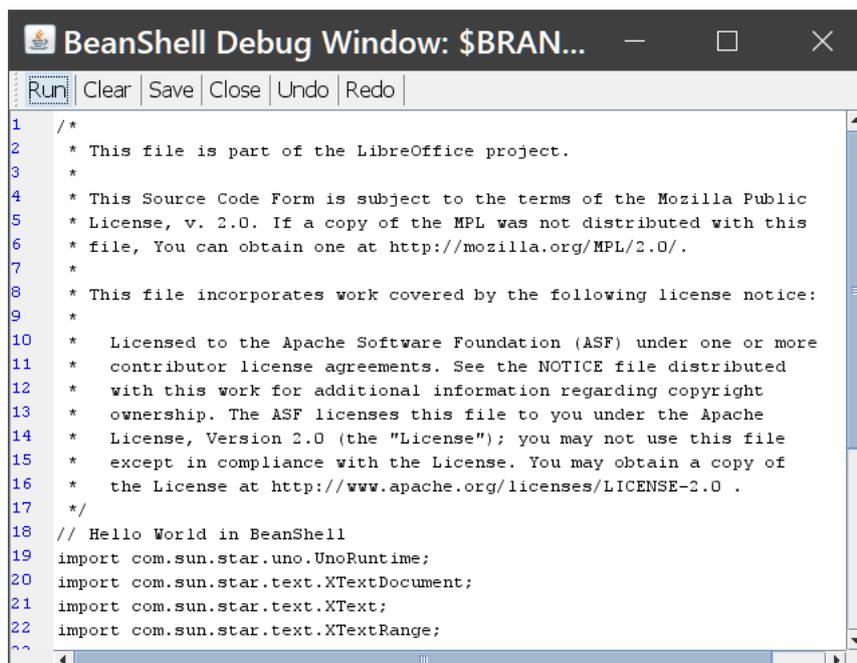


Figure 13 : la boîte de dialogue *BeanShell Debug Window*.

Le Listing 7 est un exemple de macro BeanShell qui insère le texte « Hello World from BeanShell » dans la cellule A1 du classeur Calc actif.

Listing 7 Exemple de macro BeanShell

```
import com.sun.star.uno.UnoRuntime;
import com.sun.star.sheet.XSpreadsheetView;
import com.sun.star.text.XText;
model = XSCRIPTCONTEXT.getDocument();

controller = model.getCurrentController();
view = UnoRuntime.queryInterface(XSpreadsheetView.class, controller);
sheet = view.getActiveSheet();
cell = sheet.getCellByPosition(0, 0);
cellText = UnoRuntime.queryInterface(XText.class, cell);
textCursor = cellText.createTextCursor();
cellText.insertString(textCursor, "Hello World from BeanShell", true);
return 0;
```

Macros JavaScript

JavaScript est un langage de script de haut niveau réalisé en 1995.

Quand vous sélectionnez **Outils > Macros > Gérer les macros > JavaScript** dans la barre de menu, Calc ouvre la boîte de dialogue *Macros JavaScript* (Figure 14).

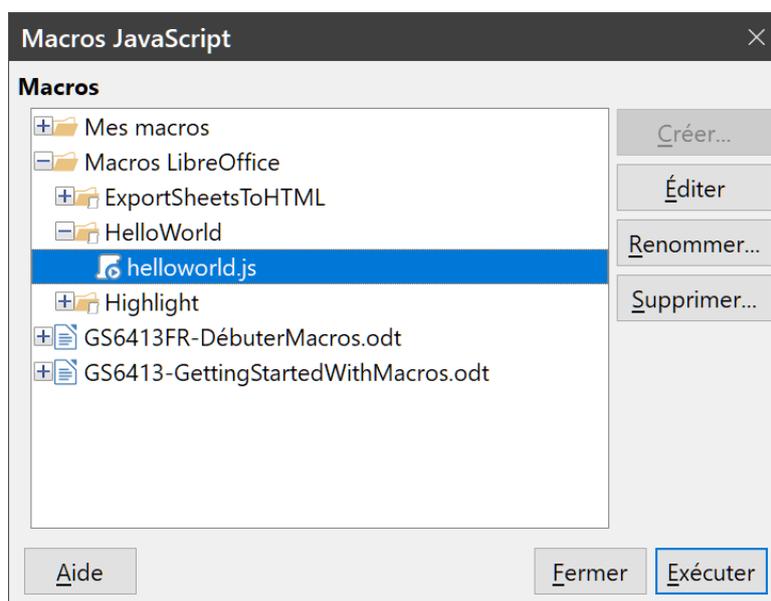


Figure 14 : la boîte de dialogue *Macros JavaScript*.

Cliquez sur le bouton **Éditer** de la boîte de dialogue *Macros JavaScript* pour ouvrir le débogueur JavaScript Rhino (Rhino JavaScript Debugger) (Figure 15). Vous pouvez trouver sur le site web de Mozilla, à l'adresse <https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino/Debugger>, des instructions détaillées sur l'utilisation de cet outil.

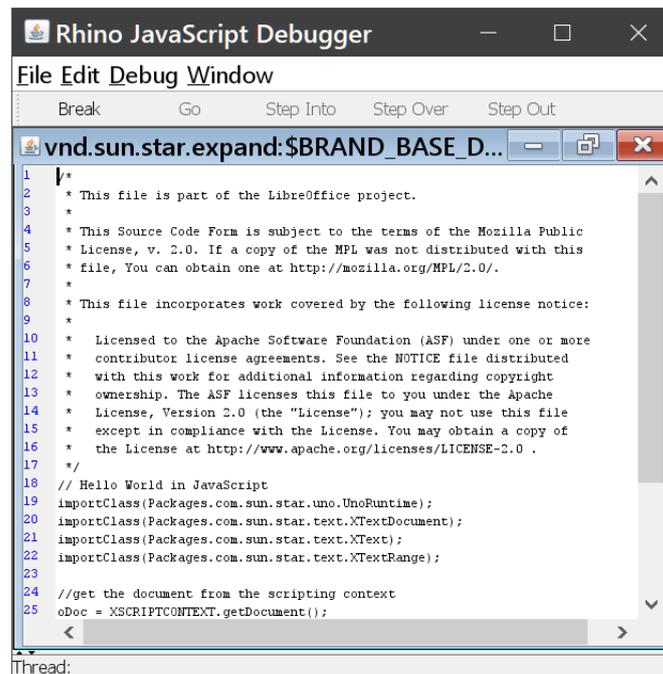


Figure 15 : le débogueur JavaScript Rhino.

Le Listing 8 propose un exemple de macro en JavaScript qui insère le texte « Hello World from JavaScript » dans la cellule A1 de la première feuille d'un classeur Calc.

Listing 8 : exemple de macro en JavaScript.

```
importClass(Packages.com.sun.star.uno.UnoRuntime);
importClass(Packages.com.sun.star.sheet.XSpreadsheetDocument);
importClass(Packages.com.sun.star.container.XIndexAccess);
importClass(Packages.com.sun.star.table.XCellRange);
importClass(Packages.com.sun.star.table.XCell);

documentRef = XSCRIPTCONTEXT.getDocument();
spreadsheetInterface = UnoRuntime.queryInterface(XSpreadsheetDocument,
documentRef);
allSheets = UnoRuntime.queryInterface(XIndexAccess,
spreadsheetInterface.getSheets());
theSheet = allSheets.getByIndex(0);
Cells = UnoRuntime.queryInterface(XCellRange, theSheet);
cellA1 = Cells.getCellByPosition(0,0);
theCell = UnoRuntime.queryInterface(XCell, cellA1);
theCell.setFormula("Hello World from JavaScript");
```

Macros Python

Python est un langage de programmation de haut niveau à usage général qui est sorti en 1991.

Quand vous sélectionnez **Outils > Macros > Gérer les macros > Python** dans la barre de menu, Calc ouvre la boîte de dialogue *Macros Python* (Figure 16).

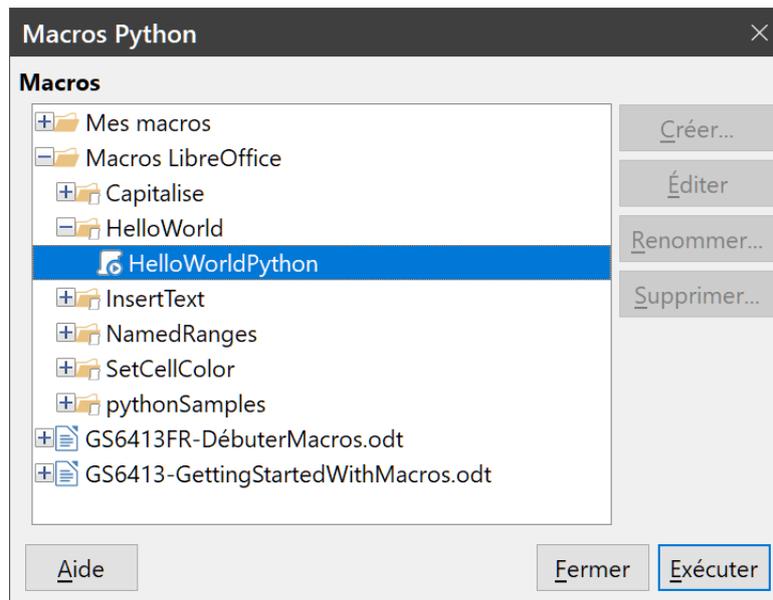


Figure 16 : la boîte de dialogue Macros Python.

Aucune fonction d'édition et de débogage des scripts Python n'est actuellement intégrée à l'interface utilisateur standard de LibreOffice. Vous pouvez cependant les modifier avec votre éditeur de texte préféré ou un EDI externe. L'extension *Alternative Python Script Organizer* (APSO) (<https://extensions.libreoffice.org/extensions/apso-alternative-script-organizer-for-python>) facilite l'édition des scripts Python, en particulier ceux qui sont incorporés à un document. Elle vous permet de configurer votre éditeur de code source préféré, de lancer l'environnement Python intégré et de déboguer les scripts Python. Si vous souhaitez plus d'informations, recherchez le terme « Python » dans le système d'aide de LibreOffice et visitez la section *Designing & Developing Python Applications* du wiki de *The Document Foundation* à l'adresse https://wiki.documentfoundation.org/Macros/Python_Design_Guide/fr.

Le Listing 9 est un exemple de macro en Python qui place le texte « Hello World from Python » dans la cellule A1 de la première feuille d'un classeur Calc.

Listing 9 : exemple de macro en Python

```
import uno

def HelloWorld():
    doc = XSCRIPTCONTEXT.getDocument()
    cell = doc.Sheets[0]['A1']
    cell.setString('Hello World from Python')
    return
```

Où trouver plus d'informations

De nombreuses ressources offrent de l'aide pour l'écriture de macros. Sélectionnez **Aide > Aide de LibreOffice** ou appuyez sur la touche *F1* pour ouvrir les pages d'aide de l'application. Le coin supérieur gauche de la fenêtre d'aide propose une liste déroulante **Module** où vous pouvez choisir l'ensemble d'informations à afficher. Pour voir l'aide du Basic, sélectionnez cette entrée tout en bas de la liste.

Matériel inclus

De nombreuses et excellentes macros sont incluses dans LibreOffice. Sélectionnez **Outils > Macros > Gérer les macros > Basic** pour ouvrir la boîte de dialogue *Macros Basic* (Figure 1). Développez la bibliothèque *Tools* dans le conteneur *Macros LibreOffice*. Observez le module *Debug* : entre autres bons exemples, vous trouverez *WritedbgiInfo* et *printdbgInfo*.

Ressources en ligne

Les liens et références suivants offrent des informations sur la programmation des macros :

<https://wiki.documentfoundation.org/Macros/fr>

<https://ask.libreoffice.org/fr/questions/> (site de questions/réponses où des bénévoles répondent aux questions concernant LibreOffice).

<https://forum.openoffice.org/fr/forum/> (forum de la communauté Apache OpenOffice où des bénévoles répondent aussi aux questions concernant LibreOffice).

https://wiki.documentfoundation.org/Documentation/Other_Documentation_and_Resources en anglais (recherchez, dans la section Programmers, le BASIC Programmer's Guide et le Developer's Guide qui contient une explication détaillée).