

Aide-mémoire LibreOffice

LibreOffice Basic

Les dialogues

v. 2.0 - 06/04/2021

Avancé

Rédigé avec LibreOffice v. 7.0.5 - Plateforme : Toutes - Jeu d'icônes par défaut (Colibre)

Dialogues Basic

Afficher un message simple

Print "Bonjour tout le monde !"

Le choix Annuler provoque l'arrêt du programme.



Afficher un message d'information

MsgBox(TexteDuMessage[, CodeDialogue[, Titre]])

Retours à la ligne dans TexteDuMessage par Chr(10) ou Chr(13).



Afficher un message et lire la réponse

Reponse = MsgBox(TexteDuMessage[, CodeDialogue[, Titre]])
ou

Reponse est un entier, selon le choix de l'utilisateur.

CodeDialogue : somme des codes boutons à afficher + icône + bouton par défaut

Boutons à afficher

0	OK	3	Oui, Non, Annuler
1	OK, Annuler	4	Oui, Non
2	Interrompre, Réessayer, Ignorer	5	Réessayer, Annuler

Icone

0	(aucun)	48	⚠ Avertissement
16	✖ Message critique	64	i Information (OK seul)
32	❓ Question		

Bouton par défaut

0	premier	256	deuxième	512	dernier
---	---------	-----	----------	-----	---------

Valeurs de retour (choix de l'utilisateur)

1	OK	3	Interrompre	5	Ignorer	7	Non
2	Annuler	4	Réessayer	6	Oui		

Fonction InputBox()

Fonction InputBox(Invite[, Titre[, ValeurDefaut])
retourne une chaîne. Si abandon, la chaîne est vide.

Dialogues de l'API

Les types FilePicker et FolderPicker ci-dessous sont influencés par Outils > Options > LibreOffice > Général. Utiliser les boîtes de dialogue LibreOffice.

Types de dialogues proposés par l'API

Sélection de fichier : objets FilePicker

com.sun.star.ui.dialogs.FilePicker	Selon option ci-dessus.
com.sun.star.ui.dialogs.OfficeFilePicker	Force le style LibreOffice.
com.sun.star.ui.dialogs.SystemFilePicker	Force le style de l'OS.

Sélection de répertoire : objets FolderPicker

com.sun.star.ui.dialogs.FolderPicker	Selon option ci-dessus.
com.sun.star.ui.dialogs.OfficeFolderPicker	Force le style LibreOffice.
com.sun.star.ui.dialogs.SystemFolderPicker	Force le style de l'OS.

L'objet FilePicker (ou OfficeFilePicker ou SystemFilePicker)

oFilePicker = CreateUnoService("com.sun.star.ui.dialogs.FilePicker")
AppendFilter() Par paires: appendFilter("NomLittéral", "*.xyz")
Ex: oFilePicker.appendFilter("Documents ODF", "*.odt;*.ods")

CurrentFilter Le filtre par défaut, parmi les filtres ajoutés par AppendFilter (nom littéral) ou le filtre choisi par l'utilisateur.

DefaultName Nom par défaut pour le fichier à enregistrer.

DisplayDirectory Le répertoire initial ou choisi par l'utilisateur.

Execute Transfère le flux d'exécution au dialogue et fournit un code de retour (voir constantes codes de retour)

Files Le tableau des fichiers sélectionnés.

initialize() Choix du type de dialogue (voir les constantes de type).

Dim FType(0) As Integer
FType(0) = 'la constante de type (ci-dessous)
oFilePicker.initialize(FType())

MultiSelectionMode (Dés)Active le mode multi-sélection (False par défaut).

Title Le titre de la fenêtre.

Constantes de types de FilePicker :

com.sun.star.ui.dialogs.TemplateDescription.XXX :	
FILEOPEN_SIMPLE	0 Dialogue d'ouverture simple.
FILESAVE_SIMPLE	1 Dialogue d'enregistrement simple.
FILESAVE_AUTOEXTENSION_PASSWORD	2 Dialogue d'enreg. : extension automatique + mot de passe.
FILESAVE_AUTOEXTENSION_PASSWORD_FILTER_OPTIONS	3 Dialogue d'enreg. enrichi : extension auto + mot de passe + options de filtrage.
FILESAVE_AUTOEXTENSION_SELECTION	4 Dialogue d'enregistrement enrichi : extension automatique + sélection.
FILESAVE_AUTOEXTENSION_TEMPLATE	5 Dialogue d'enregistrement enrichi : extension automatique + liste « Modèles ».
FILEOPEN_LINK_PREVIEW_IMAGE_TEMPLATE	6 Dialogue d'ouverture enrichi : insertion comme lien + prévisualisation + modèle.
FILEOPEN_PLAY	7 Dialogue d'ouverture enrichi : jouer.
FILEOPEN_READONLY_VERSION	8 Dialogue d'ouv. : lecture seule + version.
FILEOPEN_LINK_PREVIEW	9 Dialogue ouverture enrichi : lien + prévisu.
FILESAVE_AUTOEXTENSION	10 Dialogue d'enreg. enrichi : extension auto.

FILEOPEN_PREVIEW
FILEOPEN_LINK_PLAY

11 Dialogue d'ouverture enrichi : prévisu.
12 Dialogue d'ouverture enrichi : insertion comme lien + jouer.

Constantes codes de retour :

com.sun.star.ui.dialogs.ExecutableDialogResults.XXX
CANCEL 0 Annulé OK 1 Validé

L'objet FolderPicker (ou OfficeFolderPicker ou SystemFolderPicker)

oFldrPicker = CreateUnoService("com.sun.star.ui.dialogs.FolderPicker")
Description Aide sur le dialogue. Sur un OfficeFolderPicker, ne fait rien.

DisplayDirectory Répertoire initial.
execute Transfère le flux d'exécution au dialogue et fournit un code de retour (voir constantes codes de retour).

Title Le titre du dialogue.

Directory Le choix de l'utilisateur (format URL).

Ouvrir un fichier unique (FilePicker)

1. Créez un FilePicker. Le type par défaut convient en général (FILEOPEN_SIMPLE),
2. initialisez-le (propriétés et méthodes ci-dessus),
3. exécutez-le,
4. au retour, lisez les choix de l'utilisateur dans les propriétés CurrentFilter, DisplayDirectory et Files (vecteur) (seul Files(0) contient une valeur).

```
Dim oFilePicker As Object, NomFichier As String
NomFichier = ""
'initialisation du FilePicker
oFilePicker = CreateUnoService("com.sun.star.ui.dialogs.FilePicker")
oFilePicker.DisplayDirectory = pRepDepart
oFilePicker.appendFilter("Classeurs Calc", "*.ods")
oFilePicker.CurrentFilter = "Classeurs Calc"
oFilePicker.Title = "Choisissez le classeur"
'exécution et vérification du retour (OK)
If oFilePicker.execute = _
    com.sun.star.ui.dialogs.ExecutableDialogResults.OK Then
    NomFichier = oFilePicker.Files(0)
End If
```

Ouvrir plusieurs fichiers (FilePicker)

1. Comme pour le précédent,
2. initialisez-le (en part. avec MultiSelectionMode = True),
3. exécutez-le,
4. au retour, le vecteur Files() contient les choix de l'utilisateur.

Enregistrer un fichier (FilePicker)

1. Créez un FilePicker,
2. initialisez-le (type FILESAVE_XXX) (et propriétés et méthodes ci-dessus),
3. exécutez-le,
4. au retour, lisez les choix de l'utilisateur dans les propriétés CurrentFilter, DisplayDirectory et Files (vecteur) (seul Files(0) contient une valeur).

Choisir un répertoire (FolderPicker)

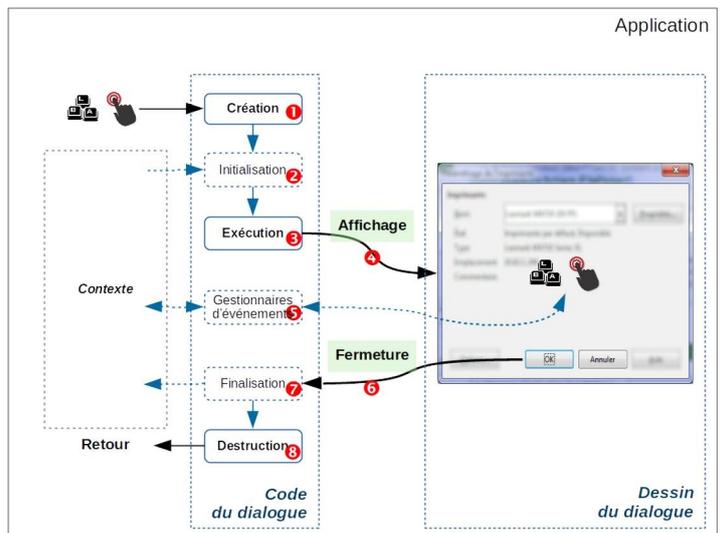
1. Créez un FolderPicker,
2. initialisez-le (propriétés et méthodes ci-dessus),
3. exécutez-le,
4. au retour, lisez le choix de l'utilisateur dans Directory.

```
Dim oFP As Object, NomRep As String
NomRep = ""
oFP = CreateUnoService("com.sun.star.ui.dialogs.FolderPicker")
oFP.DisplayDirectory = ConvertToURL("C:\Chem\Vers\Repertoire")
oFP.Description = "Cliquez sur un répertoire"
oFP.Title = "Choisissez votre répertoire de sauvegarde"
If oFP.execute = _
    com.sun.star.ui.dialogs.ExecutableDialogResults.OK Then
    NomRep = oFP.Directory
End If
```

Dialogues personnalisés – Principes

Dialogue Basic = un module Dialogue (dessin) + (au moins) un module de code.

Séquence d'exécution d'un dialogue



La séquence d'exécution d'un dialogue est illustrée ci-dessus :

1. En réponse à un événement de l'application, vous créez le dialogue ;
2. (réalisez l'initialisation des contrôles depuis le contexte applicatif si nécessaire) ;
3. exécutez le dialogue, qui reçoit le flux d'exécution :
 4. affichage,
 5. (gestion des événements sur les contrôles du dialogue),
 6. des événements provoquent la fermeture du dialogue (OK, Annuler) ;
7. (réalisez les opérations de finalisation vers le contexte applicatif si nécessaire) ;

8. le dialogue est **détruit** et **retour** à l'application appelante.

☞ Création, initialisation, exécution, finalisation et destruction : dans votre code.

Affichage, fermeture : opérations automatiques consécutives aux précédentes.

☞ Prévoyez les réponses aux **événements** sur les contrôles (aide-mémoire n°4).

Chargement des bibliothèques de dialogues

☞ Si beaucoup de code ou si plusieurs dialogues, pensez à les héberger dans des bibliothèques dédiées (une par dialogue).

☞ Les bibliothèques de dialogues ne sont jamais chargées automatiquement.

☞ Noms des bibliothèques à charger : respectez la casse !

Modal vs Non modal

Modal Un dialogue modal prend le contrôle total de clavier-souris-écran dans l'attente d'une action de l'utilisateur. L'application sous-jacente devient inaccessible.

☞ Par défaut les dialogues sont modaux.

Non modal Un dialogue non modal ne bloque pas l'accès à l'application.

Ex : le dialogue **Rechercher & remplacer** dans LibreOffice

☞ Attention aux exécutions multiples des dialogues non modaux qui peuvent bloquer l'application.

Dialogues personnalisés ordinaires (modaux)

(Cas le plus fréquent) Soient un module de dialogue `MonDlg` et un module de code `MonDlgCode` dans la bibliothèque `MaBibliDlg`. Dans une `Sub` du module de code, on instancie un objet dialogue (`oDlg`) à partir du dialogue.

Création / chargement en mémoire

```
DialogLibraries.LoadLibrary("MaBibliDlg")
oBibli = DialogLibraries.GetByName("MaBibliDlg")
oModule = oBibli.GetByName("MonDlg")
oDlg = CreateUnoDialog(oModule)
'on manipule ensuite oDlg
```

Appel seul

`oDlg.execute` ☞ Le flux d'exécution est transféré au dialogue. Pas de contrôle de validation.

Appel et test du retour

```
If oDlg.execute = com.sun.star.ui.dialogs.ExecutableDialogResults.OK
Then ...
```

☞ Le flux d'exécution est transféré au dialogue et la valeur de retour testée (ex : l'utilisateur a-t-il cliqué OK ?).

Terminaison / destruction du dialogue

`oDlg.dispose`

Exemple récapitulatif (module de code)

Cet exemple ne montre pas de gestion d'événement.

```
Sub AfficherDialogue()
Dim oBibli As Object, oModule As Object, oDlg As Object
DialogLibraries.LoadLibrary("MaBibliDlg")
oBibli = DialogLibraries.GetByName("MaBibliDlg")
oModule = oBibli.GetByName("MonDlg")
oDlg = CreateUnoDialog(oModule)
'initialiserDlg() 'initialiser le contenu du dialogue si besoin
If oDlg.execute = com.sun.star.ui.dialogs.ExecutableDialogResults.OK
Then
'FinaliserDlg() 'faire qqch avec les saisies si besoin
End If
oDlg.dispose
End Sub
```

Dialogues personnalisés non modaux

Soient un module de dialogue `MonDlgNM` et un module de code `MonDlgCodeNM` dans la bibliothèque `MaBibliDlgNM`. Dans une `Sub` du module de code, on instancie un objet (`oDlg`) à partir du dialogue.

Appliquez les mêmes principes que ci-dessus, avec des subtilités :

1. **L'affichage** est assuré par `oDlg.SetVisible(True)` au lieu de `oDlg.execute`,
2. deux variables globales booléennes de contrôle (drapeaux) :
 - `gEnCours` empêche les exécutions multiples,
 - `gAffiche` contrôle la présence du dialogue à l'écran,
3. **les réponses aux évts** (contrôles) mettent `gAffiche` à `False` pour fermer le dialogue.

Affichage du dialogue

`oDlg.SetVisible(True)`

☞ Le dialogue est **affiché**. Le flux d'exécution n'est **pas** transféré au dialogue.

Exemple récapitulatif (module de code)

```
Dim gAffiche As Boolean 'drapeau affichage
Dim gEnCours As Boolean 'drapeau exécution; empêche multi-exéc.

Sub AfficherDialogueNonModal()
'gère la création et l'affichage du dialogue
Dim oBibli As Object, oModule As Object, oDlg As Object
'éviter les lancements multiples
If Not gEnCours Then
gEnCours = True
gAffiche = True
DialogLibraries.LoadLibrary("MaBibliDlgNM")
oBibli = DialogLibraries.GetByName("MaBibliDlgNM")
oModule = oBibli.GetByName("MonDlgNM")
oDlg = CreateUnoDialog(oModule)
'initialiserDlg() 'initialiser le contenu du dialogue

'afficher le dialogue tant que la var. est True
Do While gAffiche
Wait 20 'permettre l'exécution d'autres logiciels
oDlg.SetVisible(True) 'maintenir à l'écran
Loop
'FinaliserDlg() 'faire qqch avec les saisies si besoin
oDlg.dispose
gEnCours = False
End If
End Sub 'AfficherDialogueNonModal

Sub OnBtnOKClick(ByRef pEvt As Object)
'Réponse à clic sur OK
'realiser les actions utiles
'finir par :
gAffiche = False '=> fin de la boucle while
'donc fermeture du dialogue
End Sub 'OnBtnOKClick
```

Associer un événement à une macro

Votre dialogue communique avec l'application via des **événements** (6 du schéma). Vous allez donc créer des macros pour répondre aux occurrences de ces événements (cf. A-M n°4) :

1. **Créez la macro** à exécuter selon le modèle ci-dessous :

```
Sub NomDeLaMacro()
End Sub
```

☞ Conseil : nommez la macro selon l'objet, l'action et le type d'événement.

```
Ex : Sub OnBoutonOKClick()
```

La `Sub` peut comporter un paramètre. Voir plus bas « Obtenir des informations... » ,

2. **sélectionnez l'objet** porteur de l'événement à intercepter,
 3. **accédez à sa configuration** (méthode variable selon l'objet),
 4. **choisissez l'événement** à intercepter,
 5. **pointez vers la macro** à exécuter sur déclenchement de l'événement (point 1).
- ☞ Plus d'informations sur les événements dans l'aide-mémoire n°4.

Obtenir des informations sur l'événement déclencheur

La macro de traitement peut interroger le paramètre entrant pour obtenir des informations complémentaires à propos de l'événement :

```
Sub ReponseEvenement(ByRef Event As Object)
End Sub
```

L'objet `Event` dépend du type d'événement qui déclenche l'appel de procédure.

Cas fréquents pour les contrôles

Pour accéder à	Interrogez
Objet contrôle appelant	<code>Event.Source</code>
Objet modèle du contrôle	<code>Event.Source.Model</code>
Objet dialogue propriétaire du contrôle	<code>Event.Source.Context</code>

Initialisation et finalisation

Ces opérations permettent les échanges entre l'application et le dialogue.

Initialisation

(2 du schéma) Le dialogue nécessite souvent des informations provenant du contexte d'exécution. La macro d'initialisation configure le dialogue à partir de ces données.

Finalisation

(7 du schéma) Il s'agit ici de réaliser l'opération inverse de la précédente : actualiser les données contextuelles à partir de celles saisies ou choisies dans le dialogue.

Gérer les modules Dialogue

LibreOffice gère les modules Dialogue indépendamment du code (A-M n°1). Il est possible de recopier ces objets d'un document vers un autre.

Recopier des modules d'une bibliothèque vers une autre

(dans le même document ou entre documents/conteneurs)

1. Dans l'IDE, ouvrez les deux documents/conteneurs source et cible,
2. ouvrez l'**Organiseur de macros Basic** (bouton ) ,
3. onglet **Boîtes de dialogue**, glissez/déposez depuis la source vers la cible.

☞ Par défaut les modules sont **déplacés**. Pour les **recopier** : **Ctrl** + glisser/déposer.

Sauvegarder une boîte de dialogue seule

1. Dans l'IDE, ouvrez le module Dialogue à sauvegarder,
2. cliquez le bouton de barre d'outils  **Exporter la boîte de dialogue**,
3. nommez le fichier et enregistrez-le.

Le document est au format XML et doté de l'extension `.xdl`.

☞ L'importation est la réciproque, via le bouton  **Importer la boîte de dialogue**.

Crédits

Auteur : Jean-François Nifenecker – jean-francois.nifenecker@laposte.net

Nous sommes comme des nains assis sur des épaules de géants. Si nous voyons plus de choses et plus lointaines qu'eux, ce n'est pas à cause de la perspicacité de notre vue, ni de notre grandeur, c'est parce que nous sommes élevés par eux. (Bernard de Chartres [attr.])

Historique

Version	Date	Commentaires
2.0	06/04/2021	Mise en forme ; MàJ pour LibO v.7.

Licence

Cet aide-mémoire est placé sous licence

Creative Commons BY-SA v3 (fr)

Informations :

<https://creativecommons.org/licenses/by-sa/3.0/fr/>

