

Aide-mémoire LibreOffice

LibreOffice Basic

Calc

v. 1.15 – 14/12/2019

Débutant

Rédigé avec LibreOffice v. 5.3.3 – Plateforme : Toutes

Documents LibreOffice

Document courant

```
Dim Doc As Object
Doc = ThisComponent
```

Ouvrir un document existant

Mode visible

```
Dim Doc As Object, CheminDoc As String
Dim Props() 'ici, ce tableau n'est pas initialisé
CheminDoc = ConvertToURL("C:\Chemin\Vers\UnClasseur.ods")
Doc = StarDesktop.loadComponentFromURL(CheminDoc, "_blank", 0, _
    Props())
```

Mode invisible

```
Dim Doc As Object, CheminDoc As String
Dim Props() As New com.sun.star.beans.PropertyValue
CheminDoc = ConvertToURL("C:\Chemin\Vers\UnClasseur.ods")
Props(0).Name = "Hidden" 'le document va être ouvert "caché"
Props(0).Value = True
Doc = StarDesktop.loadComponentFromURL(CheminDoc, "_blank", 0, _
    Props())
```

Rendre visible a posteriori

```
Doc.CurrentController.Frame.ContainerWindow.Visible = True
Doc.CurrentController.Frame.ContainerWindow.toFront()
```

Créer un nouveau document Calc

À partir (1) du modèle par défaut ou (2) d'un modèle spécifique.

```
Dim Doc As Object
Dim Props() 'ici, ce tableau n'est pas initialisé
Modele = "private:factory/scalc" '(1)
'ou bien
Modele = "C:\Chemin\Vers\UnModeleDeClasseur.ots" '(2)
Doc = StarDesktop.loadComponentFromURL(Modele, "_blank", 0, Props())
```

Enregistrer un document

Le document existe déjà

(équivalent à Fichier > Enregistrer)

Utilisez la méthode store de l'objet document. Ex: ThisComponent.store

Le document n'a pas encore été enregistré

(équivalent à Fichier > Enregistrer sous)

```
Dim Doc As Object, CheminDoc As String
Dim Props() 'les propriétés d'enregistr. (vide)
CheminDoc = convertToURL("C:\Chemin\Vers\UnClasseur.ods")
Doc.storeAsURL(CheminDoc, Props())
```

☞ Si copie, elle devient le document actif.

Enregistrer une copie

Comme ci-dessus mais avec Doc.storeToURL(CheminDoc, Props())

☞ La copie ne devient pas le document actif.

Enregistrer avec mot de passe (tous modes)

Ajoutez une propriété Password et donnez-lui la valeur requise.

```
Props(n).Name = "Password"
Props(n).Value = "mypwd"
```

Fermer un document

Utilisez la méthode close de l'objet document : ThisComponent.close(True)

Informations sur le document

L'objet document expose des propriétés

Location Le répertoire de stockage du document.
 ☞ Chaîne vide si pas encore enregistré.
 DocumentProperties (Object) Des propriétés complémentaires (ci-dessous).

DocumentProperties

Author	Nom de l'auteur.	ModifyDate	Date de la dernière modification
CreationDate	Date de création.	Subject	Rubrique Sujet (String).
Description	Rubrique Commentaires.	Title	Rubrique Titre (String).
ModifiedBy	Nom de l'utilisateur ayant modifié le document.	UserDefinedProperties	Propriétés personnalisées (Object).

Est-ce un document Calc ?

L'objet Doc pointe vers le document (ex: Doc = ThisComponent).
 CalcOK = Doc.SupportsService("com.sun.star.sheet.SpreadsheetDocument")

Calc – Fonctionnalités générales

L'objet Doc pointe vers le document (ex: Doc = ThisComponent).

Calcul automatique

Activé ? (Boolean)	Auto = Doc.isAutomaticCalculationEnabled
Désactiver	Doc.enableAutomaticCalculation(False)
Activer	Doc.enableAutomaticCalculation(True)
Forcer le recalcul	Doc.calculate (formules pas à jour seulement) Doc.calculateAll (tout)

Protéger le classeur

Classeur protégé ?	Test = Doc.isProtected
Protéger le classeur	Doc.protect(MotPasse) [vide possible]
Déprotéger le classeur	Doc.unprotect(MotPasse)

Feuilles (sheets)

L'objet Doc pointe vers le document (ex: Doc = ThisComponent).

Accéder aux feuilles

On peut travailler avec les objets feuilles (index base 0) :

La feuille active	Feuil = Doc.CurrentController.ActiveSheet
Liste des feuilles	LesFeuilles = Doc.Sheets
Nombre de feuilles	NbFeuilles = Doc.Sheets.Count
Objet feuille (par son index)	Feuil = Doc.Sheets(index)
Objet feuille (par son nom)	Feuil = Doc.Sheets.getName(NomFeuille)
Vérifier existence (nom)	Exist = Doc.Sheets.hasByName(NomFeuille)
Index d'une feuille	Index = Feuil.RangeAddress.Sheet

Modifier les feuilles

p est la position dans le classeur (base 0).

Ajouter une feuille nommée Nom	Doc.Sheets.insertNewByName(Nom, p)
Supprimer une feuille	Doc.Sheets.removeByName(NomFeuille)
Dupliquer une feuille	Doc.Sheets.copyByName(NomSource, NomCible, p)
Déplacer une feuille	Doc.Sheets.moveByName(NomFeuille, p)

Gérer les feuilles

Soit Feuil un objet feuille.

Activer une feuille	Doc.CurrentController.ActiveSheet = Feuil
Masquer/montrer une feuille	Feuil.IsVisible = False 'True
Tester la protection	Protegee = Feuil.IsProtected
Protéger une feuille	Feuil.protect(MotPasse) (MotPasse peut être vide)
Déprotéger une feuille	Feuil.unprotect(MotPasse)
Colorier l'onglet	Feuil.tabColor = RGB(255, 255, 0)

Lier une feuille

Lier à un fichier (ex: CSV)	Feuil.link(URL, "", "Text - txt - csv (StarCalc)", _ Filtre, com.sun.star.sheet.SheetLinkMode.VALUE)
Libérer un lien	Feuil.setLinkMode(com.sun.star.sheet.SheetLinkMode.NONE)

Trouver la dernière ligne/colonne utilisée

Feuil est l'objet feuille à explorer. Ligne et Col sont les informations recherchées.

```
Dim oCur As Object ' curseur sur la cellule
Dim oPlage As Object ' la plage utilisée
Dim Ligne As Long, Col As Long
oCur = Feuil.createCursorByRange(Feuil.getCellRangeByName("A1"))
oCur.gotoEndOfUsedArea(True)
oPlage = Feuil.getCellRangeByName(oCur.AbsoluteName)
Ligne = oPlage.RangeAddress.EndRow
Col = oPlage.RangeAddress.EndColumn
```

Cellules (cells)

Ci-dessous Cel est un objet cellule.

Accéder aux cellules

Soit et Feuil un objet feuille. On peut accéder à un objet cellule :

Par ses coordonnées	Cel = Feuil.getCellRangeByName("A4")
Par son nom	Cel = Feuil.getCellRangeByName("TVA")
Par ses coordonnées en X,Y	Cel = Feuil.getCellByPosition(0,3) Avec X=0 (col.A) ; Y=3 (lig.4)

Accéder à la cellule active

Soient Doc un objet document et ActiveCel l'objet cellule active recherché.

```
If Doc.currentSelection.supportsService _
    ("com.sun.star.sheet.SheetCell") Then
    'c'est une cellule
    ActiveCel = Doc.currentSelection
End If
```

Sélectionner une cellule

ThisComponent.CurrentController.select(Cel)

Coordonnées d'une cellule

Coordonnées (Object)	Coord = Cel.CellAddress
Rang de la feuille (Integer)	NumF = Cel.CellAddress.Sheet
Rang de la colonne (Long)	NumC = Cel.CellAddress.Column
Rang de la ligne (Long)	NumL = Cel.CellAddress.Row
Objet feuille conteneur	Feuil = Cel.Spreadsheet
Coordonnées absolues (String)	Coord = Cel.AbsoluteName

(Dé)Protéger des cellules

La propriété Cel.CellProtection peut prendre les valeurs booléennes :

Interdire les modifications	CellProtection.IsLocked = True
Cacher la formule	CellProtection.IsFormulaHidden = True
Cacher la cellule	CellProtection.IsHidden = True
Ne pas imprimer la cellule	CellProtection.IsPrintHidden = True

Accéder au contenu d'une cellule

Propriétés

Lire un contenu texte	MonTexte = Cel.String
Lire un contenu numérique	UnNombre = Cel.Value
Lire une formule (US)	LaFormule = Cel.Formula
Lire une formule (langue locale)	LaFormule = Cel.FormulaLocal
Connaitre le type de contenu	LeType = Cel.Type
Vider une cellule	Cel.String = ""

Types de contenus (propriété Type)

Les constantes com.sun.star.table.CellContentType.XXX permettent de connaître le type d'information que portent les cellules (Cel.Type, ci-dessus) :

EMPTY	Cellule vide	VALUE	Contenu valeur numérique
TEXT	Contenu texte	FORMULA	Contenu formule

Écrire dans une cellule

Remplacer le texte existant	Cel.String = "Coucou !"
Remplacer la valeur existante	Cel.Value = 1,234
Remplacer la formule existante	Cel.Formula = "=AND(A1="OUI";A2="OK")"
Remplacer la formule existante	Cel.FormulaLocal = "=ET(A1="OUI";A2="OK")"

Plages (ranges)

Plage = ensemble de cellules (y compris une seule) : Dim Pl As Object

Accéder aux plages

Soit Feu il un objet feuille. On récupère un objet plage Pl :

```
Par ses coordonnées Pl = Feu il.getCellRangeByName("C2:G14")
Par son nom Pl = Feu il.getCellRangeByName("NomDePlage")
Par ses coordonnées Pl = Feu il.getCellRangeByPosition(2, 1, 6, 13)
(X1, Y1, X2, Y2)
Arbitrairement Pl = ThisComponent.Sheets.getCellRangeByPosition(2,
(ex 3° feuille classeur) 1, 6, 13, 2)
```

Accéder à la plage active

Comme pour accéder à la cellule active, mais vérifiez

```
"com.sun.star.sheet.SheetCellRange" ou "[...]SheetCellRanges".
```

Sélectionner une plage

ThisComponent.CurrentController.select(MaPlage) où MaPlage est un objet.

Coordonnées d'une plage

Coordonnées (Object)	Coord = MaPlage.RangeAddress
Rang de la feuille (Integer)	Rang = MaPlage.RangeAddress.Sheet
Rang de la colonne (Long)	NumCHG = MaPlage.RangeAddress.StartColumn
angle haut/gauche	
Rang de la ligne (Long)	NumLHG = MaPlage.RangeAddress.StartRow
angle haut/gauche	
Rang de la colonne (Long)	NumCBD = MaPlage.RangeAddress.EndColumn
angle bas/droite	
Rang de la ligne (Long)	NumLBD = MaPlage.RangeAddress.EndRow
angle bas/droite	
Objet feuille conteneur	Feuil = MaPlage.Spreadsheet
Coordonnées absolues (String)	Coord = MaPlage.AbsoluteName

Plages nommées

L'objet Doc pointe vers le document. Avec Dim LesPlages As Object

Les plages nommées	LesPlages = Doc.NamedRanges
Nombre (Long)	Nb = LesPlages.Count
Accéder à une plage (par index)	MaPlage = LesPlages(index)
Vérifier existence (nom)	Existe = LesPlages.hasByName(Nom)
Accéder à une plage (par nom)	MaPlage = LesPlages.getByName(Nom)
Ajouter une plage	LesPlages.addNewByName(Nom, Coord, _
Coord : coordonnées de la plage	CellRef.CellAddress, 0)
CellRef : objet cellule de réf.	
Supprimer (par nom)	LesPlages.removeByName(Nom)

Effacer une plage

Effacer le contenu de MaPlage MaPlage.clearContents(ModeEff)
ModeEff est une valeur qui détermine le type d'effacement. Les constantes
com.sun.star.sheet.CellFlags.XXX permettent ce choix (combinaison avec +):
ANNOTATION Commentaires | STRING Texte
DATETIME Nombres au format date-heure | VALUE Nombres (sauf date-heure)
FORMULA Formules

Accéder au contenu des cellules d'une plage

MaPlage.DataArray est un tableau des valeurs des cellules pour MaPlage.

Recopier le contenu d'une plage vers une autre

Soient deux plages Source et Cible, de mêmes dimensions.
Recopier le contenu (valeurs) de Source Cible.DataArray = Source.DataArray
dans Cible.

Écrire des valeurs dans une plage

Soient MaPlage un objet plage et Tablo un tableau, de mêmes dimensions, dont les valeurs
sont à transférer dans la plage.

```
Dim Tablo As Variant
Tablo = MaPlage.DataArray 'Tablo prend les dim de la plage
!(donner des valeurs aux éléments du tableau)
MaPlage.DataArray = Tablo
```

MaPlage.DataArray est un tableau imbriqué : utilisez .DataArray(i)(j)

Parcourir les cellules d'une plage

À partir d'une collection (Plages.Cells) créez une enumeration. Celle-ci est parcourue en
appelant ses propriétés hasNextElement et NextElement :

```
Dim Plages As Object
Plages =
ThisComponent.createInstance("com.sun.star.sheet.SheetCellRanges")
Plages.insertByName("", MaPlage)
LEnum = Plages.Cells.CreateEnumeration
Do While LEnum.hasMoreElements
Cellule = LEnum.NextElement
!appliquer un traitement à partir de l'objet Cellule
Loop
```

Les cellules vides ne sont pas parcourues !

Plages : divers

Fusionner les cellules de MaPlage MaPlage.Merge

Types de plages

Selon le mode d'accès à une plage, celle-ci implémente l'un des services :

- ① com.sun.star.sheet.SheetCell
- ② com.sun.star.table.CellRange
- ③ com.sun.star.sheet.NamedRange
- ④ com.sun.star.sheet.SheetCellRange
- ⑤ com.sun.star.sheet.SheetCellRanges

En fonction du service implémenté, les plages devront être utilisées différemment.
Testez au moyen de leur méthode supportsService() (ex. ci-dessous)

Plage ou cellule ?

Pour connaître le type d'un objet, testez l'objet plage/cellule avec supportsService() :

```
If MonObj.supportsService(nom_du_service) Then ...
```

Remplacez nom_du_service par :

Cellule ? "com.sun.star.sheet.SheetCell" ①

Plage simple ? "com.sun.star.sheet.SheetCellRange" ④

Plage multiple ? "com.sun.star.sheet.SheetCellRanges" ⑤

Test cellule toujours avant plage simple car une cellule est aussi une plage simple !

Lignes/Colonnes (rows/columns)

Lignes et colonnes sont des propriétés des objets Sheet et Range.

Généralités

Lignes (Objet LesLignes)	LesLignes = MaPlage.Rows
Colonnes (Objet LesCols)	LesCols = MaPlage.Columns
Nombre	NbL = MaPlage.Rows.Count
	NbC = MaPlage.Columns.Count
Une ligne (Objet LaLigne) (base 0)	LaLigne = MaPlage.Rows(index)
Une colonne (Objet LaCol) (base 0)	LaCol = MaPlage.Columns(index)

Propriétés des lignes/colonnes

S'appliquent à des objets Ligne ou Lignes (resp. Colonne ou Colonnes).

Visible/Masquée (Boolean) IsVisible = True

Largeur optimale ou non (Boolean) OptimalWidth = True

Insérer/supprimer des lignes/colonnes

Soit un objet LouC. PremPos et Nombre sont les positions de début et le nombre de lignes
(resp. colonnes) à ajouter/supprimer (Long).

Insérer	LouC.insertByIndex(PremPos, Nombre)
Effacer	LouC.clearContents(ModeEff) [ModeEff : voir Effacer une plage]
Supprimer	LouC.removeByIndex(PremPos, Nombre)

Fixer lignes/colonnes

Uniquement sur un classeur visible.

Utiliser l'objet Contrôleur : MonContrl = ThisComponent.CurrentController

Yen a-t-il ? Fixes = MonContrl.hasFrozenPanels

Fixer (X, Y) MonContrl.freezeAtPosition(1, 2)

Supprimer MonContrl.freezeAtPosition(0, 0)

Appeler une fonction Calc

Utilisez le service "com.sun.star.sheet.FunctionAccess"

Principe

```
Dim FCalc As Object, Resultat As (selon contexte)
Dim Params As (selon contexte), NomFonction As String
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Resultat = FCalc.callFunction(NomFonction, Params)
```

Son nom, les arguments et le type du résultat dépendent de la fonction choisie.

Le nom de la fonction doit être le nom anglais de celle-ci.

Pour le connaître, basculez temporairement l'affichage des noms de fonctions dans Calc,
par Outils > Options > LibreOffice Calc > Formule, Utiliser les noms [...] anglais.

Exemple 1 (fonction SOMME())

```
Dim FCalc As Object, Resultat As Long
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Resultat = FCalc.callFunction("SUM", Array(1, 55, 321, 8))
```

Exemple 2 (fonction ARRONDI())

```
Dim FCalc As Object, Resultat As Double
Dim Params(1) As Variant
Params(0) = 1,2345 'nb à arrondir
Params(1) = 3 '3 décimales
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Resultat = FCalc.callFunction("ROUND", Params())
```

Créer une fonction Calc

Création

Exemple : calcul de la surface d'un trapèze (S = ((B + b) / 2) × H)

```
Function SurfaceTrapeze(GB As Double, PB As Double, H As Double) As Double
SurfaceTrapeze = ((GB + PB) / 2) * H
End Function
```

Utilisation dans Calc

Si A2 est la grande base, A3 la petite base et A4 la hauteur, la surface du trapèze s'obtient en insérant la formule suivante dans une cellule : =SURFACETRAPEZE (A2;A3;A4)

La macro reçoit les valeurs des arguments et non pas l'objet cellule émetteur.

La macro renvoie une valeur. Elle ne peut pas agir sur une cellule.

La fonction doit être placée dans une bibliothèque accessible au moment de son utilisation (ex : Standard du document ou de l'utilisateur) (sinon, erreur #VALEUR!)

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
1.01	01/10/2017	Première version.
1.15	14/12/2019	Ajout protection. Modifications mineures.

Licence

Cet aide-mémoire est placé sous licence

Creative Commons BY-SA v3 (fr).

Informations

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

