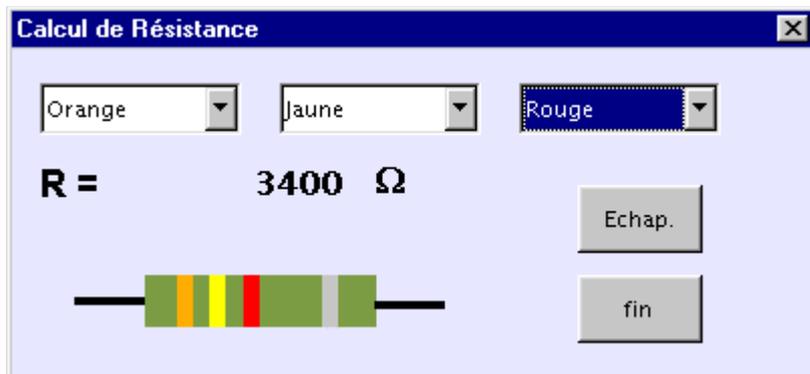


Programmation d'OpenOffice (boîtes de dialogues)

La deuxième partie de cette étude va porter sur la réalisation d'une boîte de dialogue permettant d'introduire dans la cellule sélectionnée la valeur d'une résistance à partir de ses couleurs :



Cette boîte pourra être appelée à l'aide d'un bouton et permettra d'insérer la valeur de la résistance dans la cellule sélectionnée au préalable :

	A	B	C	D	E	F	
1	Choisir une cellule puis cliquer sur le bouton.						
2					Ω		
3							
4							
5							
6							

I. Préparation.

Code des couleurs :

noir	0
marron	1
rouge	2
orange	3
jaune	4
vert	5
bleu	6
violet	7
gris	8
blanc	9

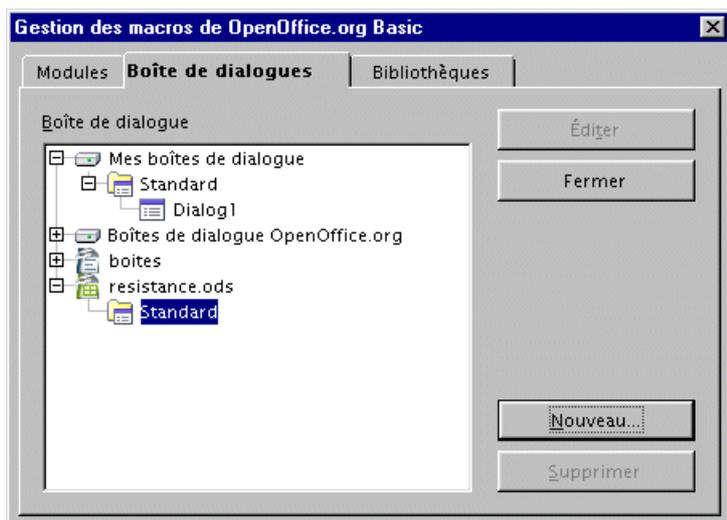
Sachant que les deux premières couleurs (C_1 et C_2) représentent deux chiffres et que la troisième (C_3) représente une puissance de dix (ex $32 \cdot 10^6$ si orange rouge bleu), proposer une formule donnant R à partir de C_1 , C_2 et C_3 .

Solution : $R = (10 \cdot C_1 + C_2) \cdot 10^{C_3}$

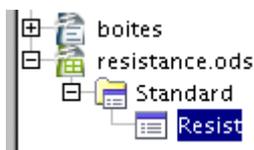
II. Construction du dialogue.

Créer un document « Classeur » dans OpenOffice et l'enregistrer sous le nom de **resistance** (en réalité le nom n'a pas d'importance mais il y sera fait référence plus loin et la compréhension s'en trouvera améliorée ainsi).

Faire **Outils/Macros/Gérer les boîtes de dialogue** puis dérouler l'arborescence liée au fichier **resistance.ods** et sélectionner **Standard**.

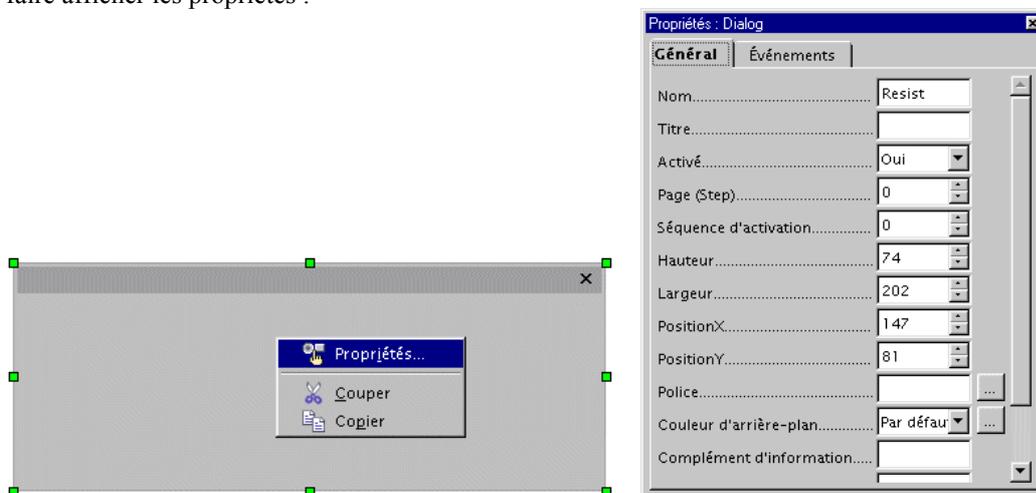


Cliquer sur le bouton nommé **Nouveau** et donner un nom au « dialogue » (je l'appellerai **Resist** dans la suite) : l'intitulé de cette boîte apparaît ainsi dans l'arborescence,



Choisir ensuite le bouton **Editer**

Modifier la taille de cette boîte de dialogue (modèle ci-dessus dans l'introduction), puis, d'un clic droit sur cette boîte, faire afficher les propriétés :



Remplir la zone **Titre** (Calcul de résistances) cela affecte le bandeau de notre boîte :



Maintenant on va mettre des listes, images et boutons dans cette boîte. Pour cela, on dispose de la barre d'outils nommée **Boîte à outils**.



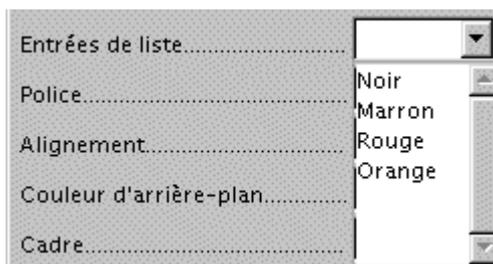
Choisir **Zone de liste**  et dessiner un rectangle :



Le panneau des propriétés contient maintenant celles de cette zone de liste (*on a ainsi accès aux propriétés d'un objet en le sélectionnant*). Changer son nom en **LB1** (c'est plus simple que ListBox1 !), puis descendre jusqu'à la propriété **Déroulante** et choisir **Oui**, et dans **Nombre de lignes** choisir **10**. Ainsi l'utilisateur aura facilement accès aux 10 couleurs lorsqu'il déroulera cette liste.



Puis entrer les couleurs dans **Entrées de liste**, la passage à la suivante se faisant en combinant au clavier **Majuscule Entrée**.



De la même façon, ajouter un bouton  et lui affecter le **Titre** « Fin » et la propriété **Type de bouton** « OK », ainsi, un clic sur ce bouton fermera le dialogue.



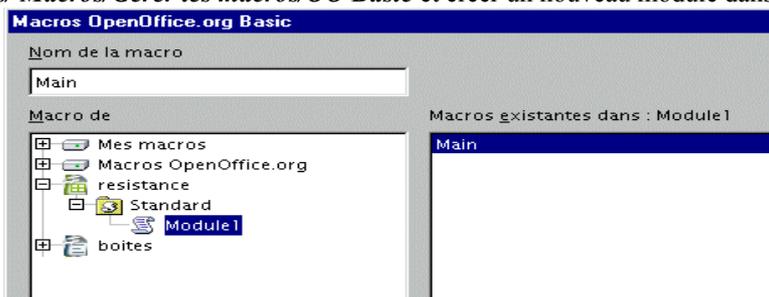
On peut effectuer un premier test en cliquant sur  , le dialogue apparaît, la liste des couleurs peut être déroulée et le bouton « Fin » ferme le tout.



Les deux autres listes déroulantes peuvent être créées par simple copier-coller, les renommer **LB2** et **LB3**. On va aussi ajouter un **Champ d'étiquette** où la valeur de R viendra s'écrire (le nommer **Lab1**).

II Ecriture du code de programmation.

Exécuter **Outils/ Macros/Gérer les macros/OO Basic** et créer un nouveau module dans « Resistance » :



Cliquer sur le bouton **Editer** pour faire apparaître l'éditeur de code.

Ecrire avant toute chose les lignes suivantes qui permettent de charger un dialogue :

```
Function LoadDialog(LibName as String, DialogName as String)
Dim oLib as Object
Dim oLibDialog as Object
Dim oRuntimeDialog as Object
oLibContainer = DialogLibraries
oLibContainer.LoadLibrary(LibName)
oLib = oLibContainer.GetByName(LibName)
oLibDialog = oLib.GetByName( DialogName)
oRuntimeDialog = CreateUnoDialog( oLibDialog)
LoadDialog() = oRuntimeDialog
End Function
```

On entre maintenant dans le vif du sujet : les variables correspondant aux listes déroulantes et au champ d'étiquette doivent être définies en dehors de tout programme pour être accessibles de tous (ce sont des variables **globales**) :

```
Dim oDialog,oLB1, oLB2, oLB3, oLab as Object
```

Le programme lancé à l'ouverture du dialogue (il est nommé ici **Main** doit affecter ces variables globales aux objets concernés : le dialogue évidemment en premier, ses composants ensuite, de plus on fait écrire « 0 » pour la valeur de la résistance :

```
Sub Main
oDialog = LoadDialog("Standard", "resist")
oLB1 = oDialog.getControl("LB1")
oLB2 = oDialog.getControl("LB2")
oLB3 = oDialog.getControl("LB3")
oLab = oDialog.getControl("Lab1")
oLab.text = "0"
oDialog.Execute()
End Sub
```

La dernière ligne commande l'exécution du dialogue. **Test** : retourner au tableur, lancer la macro **Main** et voir si cela fonctionne.

Il faut maintenant que lorsqu'on choisit une couleur, la valeur de R soit mise à jour :

```
sub maj
dim x1, x2, x3, x as double

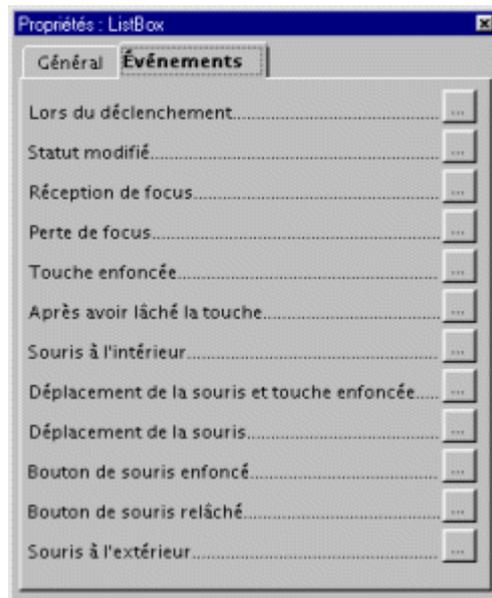
x1 = oLB1.SelectedItemPos
if x1 < 0 then x1 = 0
x2 = oLB2.SelectedItemPos
if x2 < 0 then x2 = 0
x3 = oLB3.SelectedItemPos
if x3 < 0 then x3 = 0
x = (x1*10+x2)*10^x3
oLab.Text = x
end sub
```

Dans ce programme, des variable x1,x2,x3 et x sont définies, puis on leur affecte le numéro de l'item sélectionné dans les listes déroulantes (ce qui est pratique ici puisque l'item de rang 0 est noir et correspond précisément au chiffre 0, marron pour le un et ainsi de suite !). Ensuite le calcul est fait et enfin, le résultat est affecté au champ d'étiquette.

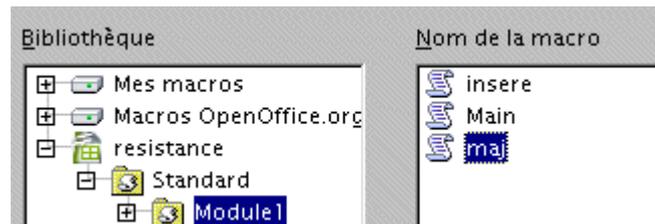
On remarque au passage que le langage basique permet facilement le transtypage puisqu'ici une valeur réelle (**double**) prend des valeurs entières et est affectée à un texte !

Il faut, pour que ces programmes soient exécutés, les affecter aux objets contenus dans le dialogue :

Sélectionner la liste déroulante **LB1** et dans la fenêtre des **Propriétés** cliquer sur l'onglet **Événements** :



Cliquer sur le petit bouton sur la ligne **Bouton de souris relâché** puis sur **Assigner** et choisir **maj** pour effectuer cette mise à jour :



Faire la même chose pour **LB2** et **LB3**. Enfin, assigner de même la macro **insere** au bouton **Fin**.

Tester le dialogue comme précédemment (macro « Main » dans le tableur). Il doit maintenant fonctionner !

III Améliorations.

1. Présentation du résultat

Il faut encadrer le champ d'étiquette des symboles **R=** et **Ω**. Il suffit de créer deux nouveaux champs d'étiquette, de leur donner comme **Titre** « R= » et « W » puis de modifier leur propriété **Police**. (Pour le deuxième on choisira la police **Symbol** pour que W devienne Ω.

Il peut aussi être bien de modifier la **couleur d'arrière plan** des champs d'étiquette pour qu'ils se « fondent » dans la boîte dialogue.

2. Dessin de la résistance

On va utiliser ici des champs d'étiquette en jouant sur leur couleur d'arrière-plan. En tracer un grand de couleur quelconque : ce sera la « résistance ». Puis quatre autres plus fins colorés en noir pour trois d'entre eux et en gris pour le quatrième : ce sont les « bandes » colorées sur la résistance. Nommer les trois bandes noires **C1**, **C2** et **C3**.

Bien entendu, la propriété **Titre** de ces champs d'étiquette est vide.

3. Modification des couleurs par programmation

Il faut définir trois variables globales pour les trois « bandes »,

Dim oDialog, oLB1, oLB2, oLB3, oLab, oC1, oC2, oC3 AS Object

les définir dans « Main » en ajoutant les lignes suivantes :

```
oC1 = oDialog.getControl("C1")
oC2 = oDialog.getControl("C2")
oC3 = oDialog.getControl("C3")
```

les mettre à jour dans « maj » :

```
select case x1
case 0
oC1.Model.BackgroundColor = &H000000
case 1
oC1.Model.BackgroundColor = &HB5850A
case 2
oC1.Model.BackgroundColor = &HFF0000
case 3
oC1.Model.BackgroundColor = &HFFAA00
case 4
oC1.Model.BackgroundColor = &HFFFF00
case 5
oC1.Model.BackgroundColor = &H00FF00
case 6
oC1.Model.BackgroundColor = &H0000FF
case 7
oC1.Model.BackgroundColor = &H8F008F
case 8
oC1.Model.BackgroundColor = &HBBBBBB
case 9
oC1.Model.BackgroundColor = &HFFFFFF
end select
```

Il faut insérer un bloc de ce type pour chaque « bande ». La variable qui contient la première couleur est, comme on l'a vu plus haut **x1** ; on effectue un test du type `select case ... end select` pour affecter la couleur correspondant à chacune des valeurs possibles à la « bande », nommée **C1**. Le nombre hexadécimal correspond aux couleurs Rouge, Vert et Bleu.

4. Insertion du résultat dans le tableur

Créer un nouveau programme nommé **insere** inspiré du code écrit lors de la création automatique d'une macro-commande et l'**assigner** à l'**enfonce**ment de la souris pour le bouton **Fin** :

```
sub insere
rem -----
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 = "StringName"
args1(0).Value = oLab.Text
dispatcher.executeDispatch(document, ".uno:EnterString", "", 0, args1())
end sub
```