

Aide mémoire Visual Basic pour Excel

Christophe Avenel¹, Kévin Huguenin², Romain Tavenard³

Pour manipuler les objets d'Excel il **faut impérativement** les stocker dans une variable préalablement définie.

Exemple :

Il est interdit de faire

```
Workbooks.open (Filename := "monFichier.xls")  
Workbooks ("monFichier.xls").maMethode (...)
```

À la place il faut utiliser

```
Dim myWorkbook as Workbook  
Set myWorkbook = Workbooks.open (Filename := "monFichier.xls")  
myWorkbook.maMethode (...)
```

En particulier, pour les zones de cellule on ne doit pas utiliser la sélection. En effet le code suivant (généré par exemple par l'enregistrement automatique de macros)

```
Range("A1").select  
Selection.Font = ...
```

, peut être remplacé par

```
Dim myRange as Range  
Set myRange = Range("A1")  
myRange.Font = ...
```

De manière générale la méthodologie à suivre pour traiter une zone de cellule est la suivante :

Définir une zone de cellule initiale

```
Dim myRange as Range  
Set myRange = Range("A1")
```

Elargir ou modifier la zone de cellule initiale de manière à obtenir la zone de cellule sur laquelle on veut travailler

```
Set myRange = myRange.currentRegion  
Set myRange = myRange.Offset(1,0)  
Set myRange = myRange.Resize(myRange.Rows.Count-1, myRange.Columns.Count)
```

Appliquer les modifications

¹ Christophe.Avenel@irisa.fr

² Kévin.Huguenin@irisa.fr

³ Romain.Tavenard@irisa.fr

```
myRange.maPropriete = ...
```

- Notez que dans l'exemple précédent, le nombre de colonnes et de lignes de la zone de cellule considérée n'est pas entré « en dur » (c'est-à-dire qu'on préférera, à des fins de généralité, se référer à `myRange.Rows.Count` plutôt qu'écrire le nombre de lignes sous forme de nombre).

Considérons maintenant les collections d'objets. Dans Visual Basic pour Excel existent trois types de collections importantes :

- `Workbooks`, la collection des classeurs ouverts dans la session Excel
- `Sheets`, la collection des feuilles d'un classeur (`Sheets` est donc une propriété de l'objet `Workbook`, et s'obtient par conséquent à l'aide d'une syntaxe du type `myWorkbook.Sheets`)
- `Range`, la collection des cellules d'une zone de cellules. Un objet de type `Range` peut donc, indifféremment, être une cellule ou un ensemble de cellules, selon l'utilisation. Par exemple, `Range("A1")` désigne la cellule A1 ou, si l'on préfère, un ensemble de cellules composé d'un seul élément : la cellule A1. Cette subtilité, qui peut paraître obscure, a un avantage certain : comme toute collection, on peut parcourir un `Range` (qu'il soit composé d'une ou plusieurs cellules) à l'aide de la boucle `For Each` :

```
Dim myRange As Range
Dim myCell As Range
Set myRange = Range("A1")
For Each myCell in myRange
    myCell.Font = ...
Next myCell
```

Pour traiter un document Excel (par exemple pour en importer les données) la méthodologie est la suivante :

- Ouvrir le document dans Excel
- Parcourir les pages de ce classeur une à une et traiter celles qui nous intéressent.
- Par exemple le code suivant effectue un traitement quelconque aux pages dont le nom commence par « ab » :

```
Dim myWorkbook as Workbook
Dim mySheet as Sheet
Set myWorkbook = Workbooks.open (Filename := "monFichier.xls")
For Each mySheet in myWorkbook.Sheets
    If (StrComp(Mid(mySheet.Name,1,3) , "ab")=0) Then
        Traiter la page ...
    End If
Next mySheet
```