Auteur	Jules FLEURY, SIGéo / CEREGE
Date de création	18/01/07
Date de dernière modification	18/01/07
Logiciels utilisés	ArcGIS, VBA, VB
Nom de la procédure	**Introduction à la programmation VBA et VB dans ArcGis**

Table des matières

A.Automatisation et développement dans ARCGIS....1
B.Introduction à Visual Basic for Applications....2
a.Utilisation de VBA dans ARCGIS....2
C.Création d'une macro dans ARCGIS....7
D.Création de contrôles pour macro dans ARCGIS....8
E.Ajout d'une dll dans arcgis....9
F.Création d'une dll dans VB....10

Programmation ArcGis

On pourra se reporter au site du P. Sawada de l'Université d'Ottawa (lien mort) dont cette partie est la traduction.
Les livres numériques livrés avec ArcGIS sont aussi très utiles et complets, et le site internet d'ESRI fournira également toutes les informations utiles.
Voir notamment : http://webhelp.esri.com/arcgiSDEsktop/9.3/index.cfm?TopicName=Creating_custom_commands_with_VBA_and_UI_Controls
et http://resources.esri.com/help/9.3/arcgisdesktop/com/COM/VBA/Working_VBA.htm

A.*Automatisation et développement dans ARCGIS*

Dans de nombreux cas, il est préférable de développer soi-même une application qui n'est pas implémentée dans le SIG : 1. pour une fonction qui n'existe pas en standard ; 2. pour une opération fastidieuse en standard ; 3. pour une opération répétitive.

ArcGIS est personnalisable à l'aide de tout langage de programmation conforme à la norme COM (par exemple, Visual Basic, Visual C++, et Visual J++). Tous les composants d'ArcGis sont utilisables pour développer des applications. Ces composants sont livrés dans la plate-forme ArcObjects. Il est possible d'étendre ArcObjects en écrivant des composants COM.
ArcView est livré avec le langage de programmation Visual Basic for Applications (VBA).
Les tâches que l'on accomplit fréquemment avec le logiciel peuvent être automatisées et de nouveaux masques, boîtes de dialogue et contrôles graphiques viennent s'ajouter aux applications ArcMap, ArcCatalog et ArcScene.

Si vous n'avez jamais programmé ARCGIS, vous pourrez pour mieux saisir la méthodologie utilisée les outils suivants dans l'ordre cité :
a. La base de la programmation dans ARCGIS se trouve dans la calculatrice de champ, qui permet de saisir des fonctions plus ou moins avancées pour le calcul d'un attribut.
b. On pourra utiliser le langage VBA (Visual Basic for Applications) qui est fournit avec ARCGIS. Cet outil, bien que pratique et puissant, ne permet pas la diffusion sur d'autres postes de manière facile.
c. On peut aussi écrire son code sur d'autres plateformes (ex : Visual Studio, Delphi, …) mais on ne bénéficie plus des variables d'entrée dans ArcMap et ArcCatalog (bien pratiques). Pour

le portage et la réutilisation d'une application, il sera souhaitable d'utiliser cette méthode qui permet de générer des Dll active X. Bien sûr il faudra disposer de l'outil de développement adéquat.

Avant de vous lancer dans le développement d'un outil, assurez vous que tout ou partie de cet outil n'existe pas déjà. Pour cela, je vous conseille d'utiliser les forums de discussion d'utilisateurs ou les sites rassemblant les scripts.

B.*Introduction à Visual Basic for Applications*

Visual Basic est un langage orienté-objet et commandé par des événements.

a.Utilisation de VBA dans ARCGIS

Nous allons présenter l'utilisation de code VBA dans ArcMap à travers la Calculatrice de champ.
Pour mettre à jour un champ, on utilise la calculatrice. Pour y accéder, faire un clic droit sur un champ puis fonction Calculer les valeurs ouvre la calculatrice.

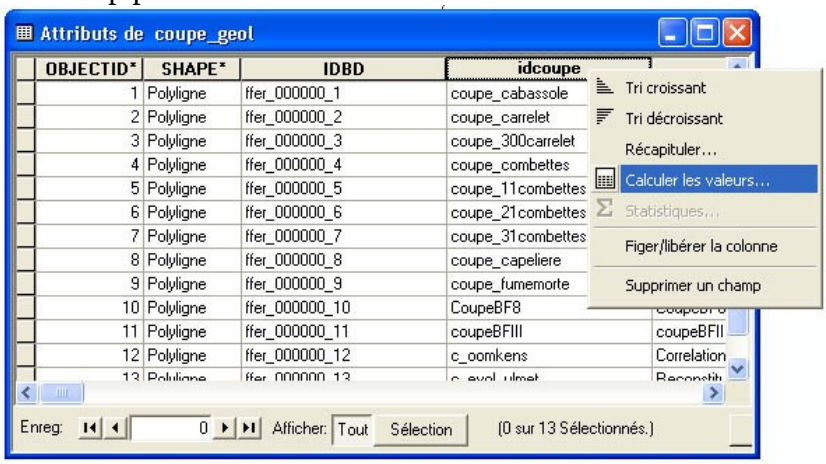


Figure 1 : Mise à jour d'un champ par la calculatrice

Une fois la calculatrice ouverte, on peut taper une expression simple (bouton avancé non coché) ou alors utiliser du code vba (bouton avancé coché) pour mettre à jour le champ :

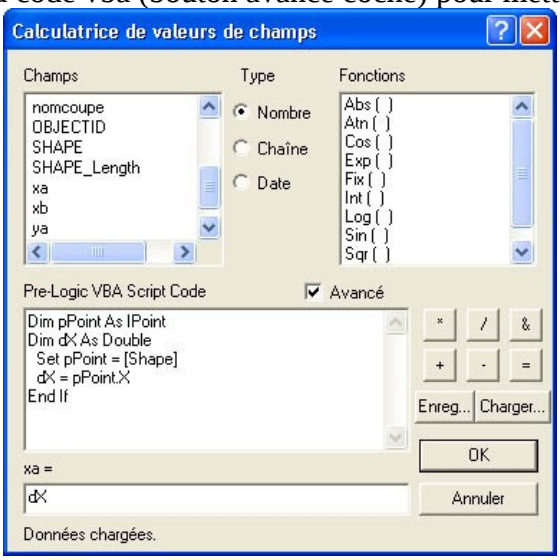


Figure 2 : Utilisation de code VBA dans la calculatrice

Par le bouton Enreg… on pourra enregistrer l'expression qui sera alors sauvegardée sous forme de fichier .cal.

Par le bouton Charger… on pourra charger des expressions au format .cal.

Calcul des coordonées de point

L'exemple ci-dessous permet le calcul de la coordonnée x de chaque point. Le code est le suivant :

Pre-Logic VBA Script Code

```
Dim dblPoint as double
Dim pPoint as IPoint
Set pPoint = [shape]
dblPoint = pPoint.X
```

[Field] =

```
dblPoint
```

Si la projection native de la classe n'est pas la même que celle de la carte et que l'on souhaite récupérer les coordonnées dans la projection de la carte, alors on utilisera le code suivant :

Pre-Logic VBA Script Code

```
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pPoint As IPoint
Dim dX As Double
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
If (Not IsNull([Shape])) Then
 Set pPoint = [Shape]
 pPoint.Project pMap.SpatialReference
 dX = pPoint.X
End If
```

[Field] =

```
dX
```

Pour calculer la coordonnée y de chaque point, utiliser la propriété Y au lieu de X.

Calcul de surface d'un polygone

Utiliser le code suivant :

Pre-Logic VBA Script Code

```
Dim dblArea as double
Dim pArea as IArea
Set pArea = [shape]
dblArea = pArea.area
```

[Field] =

```
dblArea
```

Calcul du centroïde d'un polygone

Utiliser le code suivant pour la coordonnée x du contrïde d'une polygone :

Pre-Logic VBA Script Code

```
Dim dblCentX as double
Dim pArea as IArea
Set pArea = [shape]
dblCentX = pArea.Centroid.X
```

[Field] =

```
dblCentX
```

Changer X par Y pour avoir la coordonnée y du centroïde.
Attention la défintion du centroïde utilisée par la méthode Centroid ne renvoie pas toujours un point à l'intérieur du polygone.
Pour avoir le centre de gravité du polygone programmer le sans cette fonction.

Calcul du périmètre d'un polygone

Utiliser le code suivant pour calculer le périmètre d'un polygone :

Pre-Logic VBA Script Code

```
Dim dblPerimeter as double
Dim pCurve as ICurve
Set pCurve = [shape]
dblPerimeter = pCurve.Length
```

[Field] =

```
dblPerimeter
```

Calcul de l'enveloppe minimum

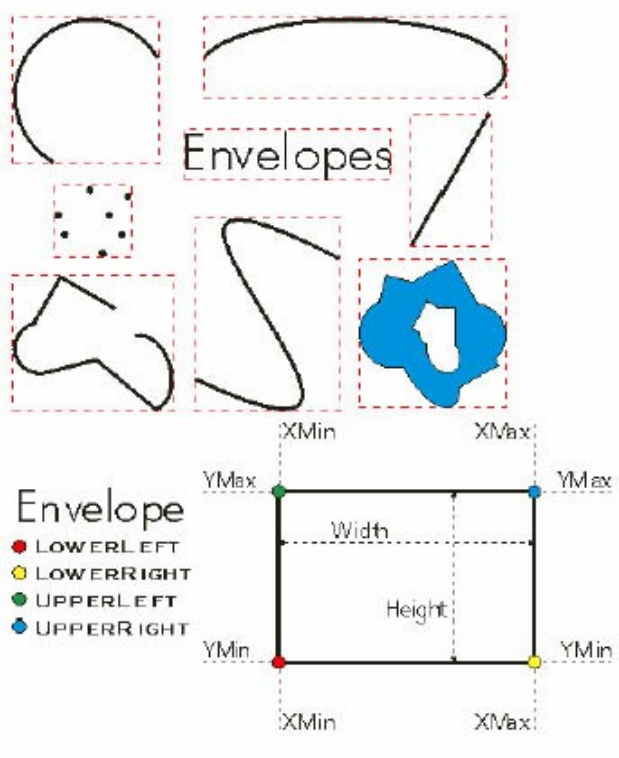


Figure 3 : Enveloppe (rectangle englobant)

On peut calculer l'enveloppe (rectangle englobant) d'une forme (polygone, polyligne, multipoint). Pour la largeur de l'enveloppe :

Pre-Logic VBA Script Code

```
dim dblGeom as Double
 dim geom as IGeometry
 set geom = [shape]
 dblGeom =geom.Envelope.width
```

[Field] = **dblGeom**

Pour la hauteur de l'enveloppe :

Pre-Logic VBA Script Code

```
dim dblGeom as Double
 dim geom as IGeometry
 set geom = [shape]
 dblGeom =geom.Envelope.Height
```

[Field] = **dblGeom**

Pour la coordonnée y maximum :

Pre-Logic VBA Script Code

```
dim dblGeom as Double
 dim geom as IGeometry
 set geom = [shape]
 dblGeom =geom.Envelope.YMax
```

[Field] = **dblGeom**

Pour la coordonnée y minimum :

Pre-Logic VBA Script Code
dim dblGeom as Double
dim geom as IGeometry
set geom = [shape]
dblGeom =geom.Envelope.YMin
[Field] = **dblGeom**

Pour la coordonnée X maximum:

Pre-Logic VBA Script Code
dim dblGeom as Double
dim geom as IGeometry
set geom = [shape]
dblGeom =geom.Envelope.XMax
[Field] = **dblGeom**

Pour la coordonnée X minimum :

Pre-Logic VBA Script Code
dim dblGeom as Double
dim geom as IGeometry
set geom = [shape]
dblGeom =geom.Envelope.XMin
[Field] =
dblGeom

Type de géométrie d'un objet

ARCGIS gère différents types de géométrie. Les plus courant étant bien sûr les polygones, les polylignes et les points mais il y en a bien d'autres :

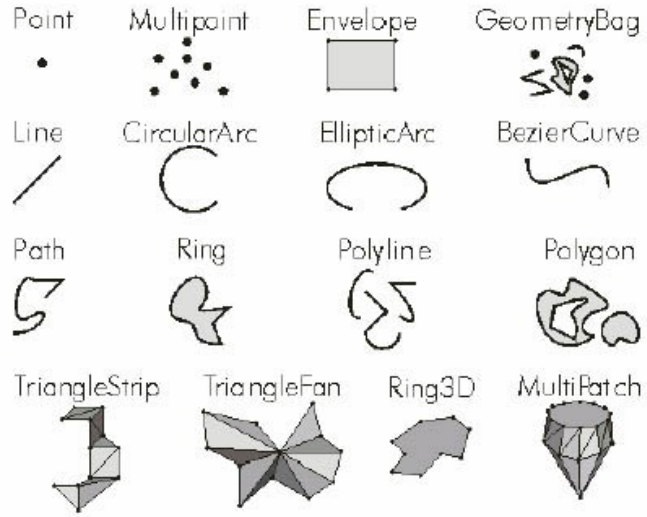


Figure 4 : Géométries gérées par ARCGIS

Ci-dessous une liste de géométries avec leur code :

esriGeometryNull = 0
esriGeometryPoint = 1
esriGeometryMultipoint = 2
esriGeometryPolyline = 3
esriGeometryPolygon = 4
esriGeometryEnvelope = 5
esriGeometryPath = 6
esriGeometryAny = 7
esriGeometryMultiPatch = 9
esriGeometryRing = 11
esriGeometryLine = 13
esriGeometryCircularArc = 14
esriGeometryBezier3Curve = 15
esriGeometryEllipticArc = 16

esriGeometryBag = 17
esriGeometryTriangleStrip = 18
esriGeometryTriangleFan = 19
esriGeometryRay = 21
esriGeometrySphere = 22

Le type de géométrie d'une classe peut être connu par la fonction :

Pre-Logic VBA Script Code

```
dim intGeomType as Integer
dim geom as IGeometry
set geom = [shape]
dblGeomType =geom.GeometryType
```

[Field] =

```
intGeomType
```

Par exemple si la valeur retournée est 4 le type de géométrie est un polygone.

Instructions conditionnelles

On peut insérer dans le code de la calculatrice des conditions if. Le type de condition peut être :

If ***[champ]*** *(=, <,>, <>, <=, >=)* ***valeur*** then
variable = nouvelle_valeur
Else
variable = nouvelle_valeur (e.g., ***[champ])***
Endif

Les signes *(=, <,>, <>, <=, >=)* sont les opérateurs.
On peut aussi utiliser la forme suivante :

If ***[fieldname]*** *(=, <,>, <>, <=, >=)* ***SomeValue*** then
output_variable = SomeNewValue
Elseif ***[fieldname]*** *(=, <,>, <>, <=, >=)* ***SomeSecondValue***
output_variable = SomeOtherNewValue (e.g., ***[fieldname])***
Elseif .....
.....
else
.....
Endif

Si l'on veut choisir entre plusieurs valeurs possibles, on pourra utiliser *select case :*

Select case ***[fieldname]***
Case ***valeur1***
instructions
Case ***valeur2***
instructions
.....
Case else
.....
end select

Comme exemple on peut vouloir classifier les superficies des polygones en 3 catégories : petite, moyenne, et grande. Pour cela on utilisera le code suivant :

Pre-Logic VBA Script Code

```
dim x as string
if [area] <=1 then
        x="petite"
elseif [area] >1 and [area] <=3 then
        x="moyenne"
elseif [area] >3 then
        x = "grande"
endif
```

[Field] =

```
x
```

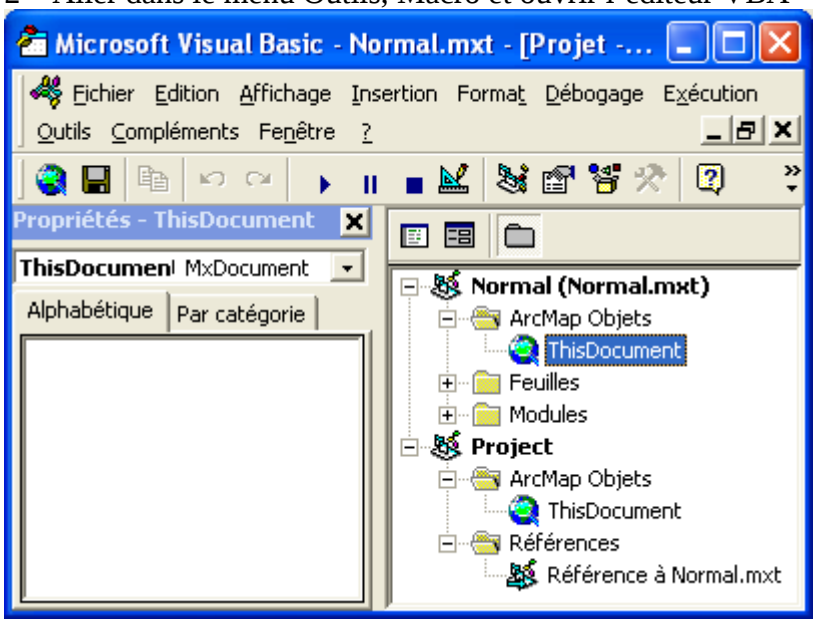
Lorsque les valeurs sont des chaînes de caractère on utilise les guillemets comme illustré ci-dessus.
Lorsque on utilise comme variable ou comme valeur un champ, alors il est entre crochets comme illustré ci-dessus par le champ [area].

C.Création d'une macro dans ARCGIS

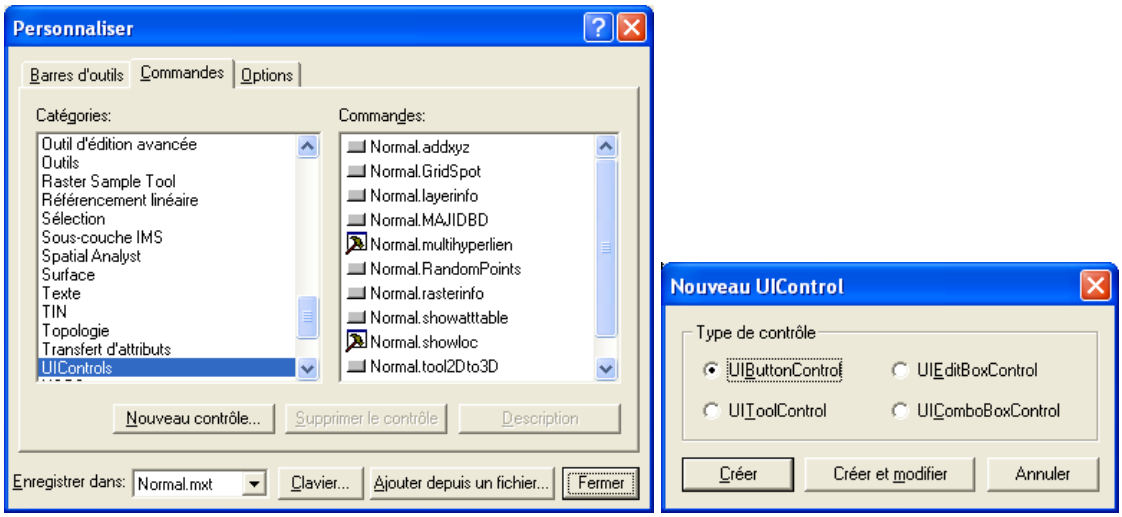
Une macro dans ArcGIS utilise le langage VBA et les composants ArcObjects pour automatiser une opération. VBA est un langage de programmation par évenements (utilisation de boutons, de formulaires qui déclenchent une opération). Avant de coder avec ArcObjects, il faudra se familiariser avec toutes les notions qu'il utilise. Citons entre autres :

- c'est une variante de la programmation objet ; on utilisera ici les termes de classe, d'objet, d'interface, de méthode et de propriété (ou attribut). Pour plus d'infos voir la documentation online fournit avec ArcGis. L'apprentissage du modèle de données d'ArcObjects est assez complexe. On pourra imprimer les diagrammes des classes pour avoir tous les objets sous les yeux.
- on entre dans le modèle de données par des variables prédéfinies : « ThisDocument » qui est le document utilisé ; « Application » qui est l'application utilisée. ArcCatalog ou ArcMap par exemple.
- On peut coder dans Normal.mxt avec ArcMap (c'est le template de tous les documents arcmap) (respectivement Normal.gxd avec ArcCatalog) auquel cas la macro sera accessible dans tous les documents. On peut coder dans Project auquel cas la macro sera accessible seulement dans le document ouvert et le code sera sauvegardé dans le document .mxd.

1- Ouvrir ArcMap
2- Aller dans le menu Outils, Macro et ouvrir l'editeur VBA



3- Vous allez saisir votre code soit dans le ThisDocument du Normal.mxt, soit dans celui du Project. Essayez d'utiliser des modules réutilisables si une partie du code peut être réutilisée dans d'autres programmes et pour simplifier la lisibilité du code.
4- sauvegarder votre projet et fermer l'editeur VBA
5- créer un contrôle pour la macro. Dans le menu Outils, Personnaliser, onglet Commandes, selectionner la catégorie UIcontrols et cliquer sur Nouveau contrôle.

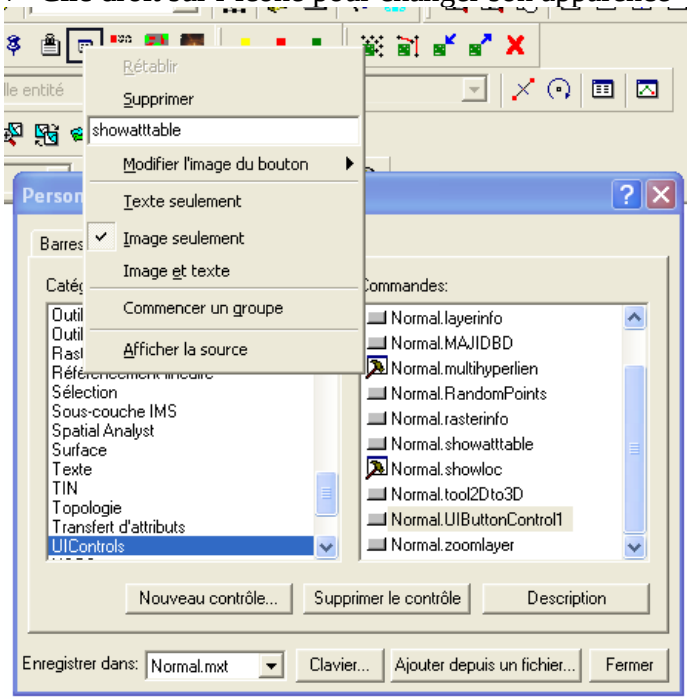


Faire Créer et vous obtenez

Le nouveau contrôle se trouve ici dans Normal.mxt, d'où le préfixe normal. Pour lui associer le code que vous venez de saisir, chagner le nom UibuttonControl1 par le nom de la procédure principale que vous avez créée.

6- Faire glisser l'icône de l'outil vers la barre d'outils où vous voulez l'insérer

7- Clic droit sur l'icône pour changer son apparence



D.*Création de contrôles pour macro dans ARCGIS*

voir le site officiel ESRI : http://webhelp.esri.com/arcgiSDEsktop/9.3/index.cfm?TopicName=Creating_custom_commands_with_VBA_and_UI_Controls
et http://resources.esri.com/help/9.3/arcgisdesktop/com/COM/VBA/Working_VBA.htm

E.*Ajout d'une dll dans arcgis*

Voir le document load dll ou le site introduction to gis du Prof. Sawada de l'Université d'Ottawa.
On pourra aussi voir le support d'ESRI France, dans la section outils et exemples de scripts d'ArcGIS Desktop.

1. Eventuellement effectuer la registration de la dll avec regsvr32, ou par un clic droit dans l'explorateur windows puis avec Register, ou en lançant le script d'inscription dans le registre qui porte le même nom que la dll.
   Pour désinstaller une dll, faire dans la boîte MsDos regsvr32 /u *nomdll*
2. Ouvrir un document dans ArcMap.
3. Exécuter la commande *Outil* > *Personnaliser* puis se placer dans la rubrique *Commandes*.
4. Dans la zone *Enregistrer dans*, sélectionner le fichier *Normal.mxt* si l'on veut que l'outil soit accessibles dans tous les documents arcmap, sinon choisir le nom d'un document dans lequel enregistrer la dll.
5. Cliquer sur le bouton *Ajouter* depuis un fichier puis sélectionner le fichier *nom.dll*.
6. Un nouvel outil est ajouté dans une des catégories de l'onglet *Commandes*.

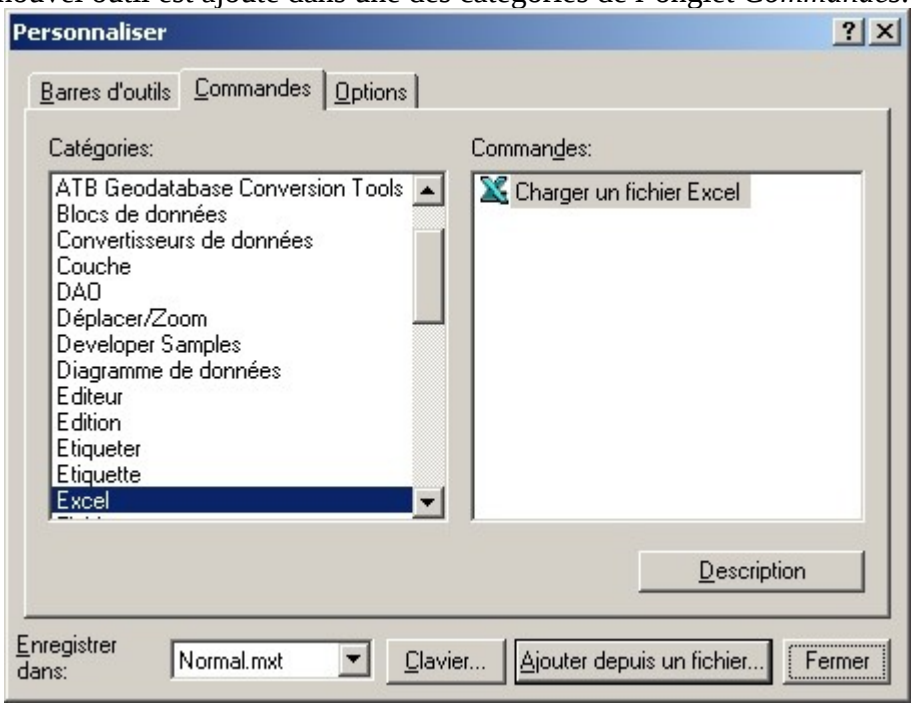


Figure 5 : Ajout d'une dll dans Normal.mxt

7. Faire glisser le nouvel outil vers une des barres de l'interface d'ArcMap.

Figure 6 : Ajout d'un bouton lié à une dll dans l'interface d'Arcmap

8. Fermer la boîte de dialogue *Personnaliser*.

F.*Création d'une dll dans VB*

Voir sur le forum d'ESRI la discussion ici.
Voici l'explication :
I.First, you need to set up the project environment in Visual Basic:

- Open VB, and click "ActiveX Dll" in the New Project dialog box.
- Since you will be using ArcObjects, you will need to reference the ESRI Object Library (esriCore.olb).
- Click on Project/References from the menu bar
- Scroll down the list and click "Esri Object Library" then "OK"
- Change the name of your project
- Click Project/Project Properties
- On the General tab, enter the new name of your project under "Project Name"
- Change the name of your class (default is Class1). You might want to name it as clsClassName (e.g., clsMyClass).
- Click F4 to bring up Properties window
- Type new name in (Name) property

- Now you are ready to start coding...(later you will cut and paste your vba code)

- At a minimum, to create any command in Visual Basic that will be used in an ESRI product, you need to implement the ICommand interface.

- If you are also creating a tool - you need to implement the ITool interface.

At the top of your code page for your class, type:

```
Option Explicit
Implements ICommand
Implements ITool

Private m_pApp as IApplication
Private m_pDoc as IMxDocument
```

- The two lines ("Implements...") will automatically populate the Object and Procedure (methods, properties) drop down boxes.
- Even if you are not going to provide code for all of the properties and methods, you have to at least "stub out" all of them. ESRI provides an Add-in that does this for you, but to do so manually, just click on ICommand in the object drop down list and click on each of the methods and properties in the procedure drop down list, then the same for ITool.

- It is a good idea to add code that at least identifies your command from others.
- For example, you can create it's own category

```
Private Property Get ICommand_Category() As String
ICommand_Category = "MyCustomCategory"
End Property
```

In addition, code should be provided for the ITool_Enabled property:

```
Private Property Get ICommand_Enabled() As Boolean
ICommand_Enabled = True
End Property
```

- You don't have access to Application like you do in VBA, so you have to get it through the hook supplied as a value passed into the ICommand's On_Create procedure.

- You already created variables for these items above, so now you need to set them:

- In the OnCreate procedure, get the hook:

```
Private Sub ICommand_OnCreate(ByVal hook As Object)

set m_pApp = hook
set m_pDoc = m_pApp.Document
```

End Sub

The set m_pDoc = m_pApp.Document in VB is the "equivalent" of set pDoc = ThisDocument in VBA.

- Go ahead and Save your project if you haven't already.
  Si vous voulez transférer la variable m_pApp à des formulaires ou des modules externs, alors déclarez dans ces formulaires (modules) une variable public papp.
  Puis dans Private Sub ICommand_OnClick(), pour un formulaire nommé frmcore (par exemple), faire
  Set Frmcore.papp = m_papp 'on établit la variable avant de charger le formulaire
  Load frmcore
  frmcore.Show

  - You are now ready to transfer your code from VBA. Just cut and paste (except for the above changes).

  - Once you have finished coding, you need to compile your code.
  - Under the File menu, choose "Make Project1 dll..." or whatever your project's name is.

  - You will be notified if there are any compilation errors.

  - Once your code compiles without errors, save it, then you are ready to use it in ArcMap.

  - Open ArcMap, and go into Tools/Customize.

  - Click on the "Commands" Tab.

  - Click "Add From File..." (You can use this as long as you only implement ICommand/ITool - if you implement any additional interfaces, you must use the Components Category - but that's another topic...)

  - Locate your .dll and click "Open"

  - Your tool appears in the category you gave it and with the name you gave it.

  - Drag it to a toolbar, and you are ready to go.