

Visual Basic pour Access

Introduction

L'utilisation du visual basic n'est pas indispensable pour la réalisation de bases de données Access. Les macros permettent de répondre à la majorité des besoins d'automatisation et sont conseillées, car plus simples à écrire, plus robustes et plus faciles à maintenir. Néanmoins l'usage de modules VBA devient indispensable dans certains cas, tels que :

- ☐ L'écriture de fonctions trop complexes pour le générateur d'expression (mathématiques, statistiques, structures alternatives imbriquées, ...)
- ☐ L'interfaçage de la base avec des applications externes (OLE)
- ☐ La gestion des transactions
- ☐ Remplacer des requêtes SQL complexes, ce qui peut permettre de gagner du temps de traitement sur des grands ensembles de données
- ☐ ...

Quelques notions de base

Les modules

Un Module est un regroupement de déclarations et de procédures Visual Basic Edition Applications, associées pour former une unité. Les déclarations et les procédures sont composées d'un code Visual Basic spécial. Il existe deux principaux types de modules : les Modules de classe et les Modules standard. Les modules de classe contiennent la définition des nouveaux objets. Les procédures définies dans ce type de modules font alors partie des propriétés et des méthodes de l'objet. Les Modules standard contiennent des procédures plus générales non associées à un autre objet et sont disponibles pour d'autres procédures dans la base de données.

Quelques rappels sur le langage Visual Basic

Déclaration de fonctions

Il existe deux type de procédures : la fonction **Function** qui renvoie une valeur. Quant à **Sub** Procédure, elle exécute une suite d'instruction sans renvoyer de valeur.

Function <Nom de procédure>

<Instructions>...

End Function

Sub <Nom de procédure>

<Instructions>

End Sub

Type de données de base

| Type de données | Signe |
|-----------------|-------|
| Integer | % |
| Long | & |
| Single | ! |
| Double | # |

| | |
|----------|----|
| Currency | @ |
| String | \$ |

ex :

Somme%=100

Dim Somme As Integer

DefInt Somme

Public X As Integer

Les tableaux

Dim Valeur(20) As Integer

Dim A As Variant

A = Array(10,20,30)

Les opérateurs

Opérateurs Arithmétiques : + , - , * , / , \ , ^ , MOD

Opérateurs de Comparaison : < , > , = , <= , >= , <> , Is , Like

Opérateurs de concaténation : + , &

Opérateurs logique : And , Eqv , Imp , Not , Or , Xor

Les branchements

If A> 10 Then A=A+1

If A>10 Then

A=A+1

B=0

End If

If A>10 Then

A=A+1

Else

B=0

End If

Select Case A

Case A<0

MsgBox (` A est inférieur à 0 `)

Case A>0

MsgBox (` A est Supérieur à 0 `)

Case Else

MsgBox (` A est égal à 0 `)

End Select

Les boucles

I= 0

For i=2 To 10 Step 2

Somme=Somme +i

Next i

Do Until i=100
i=i+1
Loop

i=0
Do While i<10
i=i+1
Loop

Do
I=i+1
Loop Until i=100

i=0
Do
i=i+1
Loop While i<10

Les objets dans Access

Chaque objet est une entité possédant ces propriétés et ces propres méthodes. Une méthode est une procédure pouvant agir sur un objet. En voici un exemple d'objets :

Application

L'application Access Actuelle

Database

Une Base de données regroupant ses propres objets (table, requêtes, formulaires, ?)

Table

Une table intégrée dans une bas de donnée

Recordset

Un enregistrement inclus dans une table

Field

Le champ d'un enregistrement