

VBA programming Visual Basic for Applications

Herve Hocquard

http://www.labri.fr/perso/hocquard



Many hard-core programmers scoff at the idea of programming in BASIC. The name itself (an acronym for Beginner's All-purpose Symbolic Instruction Code) suggests that BASIC isn't a professional language.

In fact, BASIC was first developed in the early 1960s as a way to teach programming techniques to college students. BASIC caught on quickly and is available in hundreds of dialects for many types of computers.

BASIC gained quite a bit of respectability in 1991 when Microsoft released Visual Basic for Windows.

This product made it easy for the masses to develop stand-alone applications for Windows.

Visual Basic has very little in common with early versions of BASIC, but Visual Basic is the foundation on which VBA was built.

The secret to using VBA with other applications lies in understanding the *object model* for each application.

VBA, after all, simply manipulates objects, and each product (Excel, Word, Access, PowerPoint, and so on) has its own unique object model. You can program an application by using the objects that the application exposes.

Excel's object model, for example, exposes several powerful data analysis objects, such as worksheets, charts, pivot tables, and numerous mathematical, financial, engineering, and general business functions. With VBA, you can work with these objects and develop automated procedures. While you work with VBA in Excel, you gradually build an understanding of the object model.

Warning: The object model will be confusing at first. Eventually, however, the pieces come together and all of a sudden, you realize that you've mastered it!

For the past few years, I've heard rumors that Microsoft is going to remove VBA from the Office applications and replace it with .NET (or something else). My understanding is that these rumors are unfounded. Sure, Microsoft has developed another way to automate Office applications, but VBA will be around for quite a while — at least in Excel for Windows.

Microsoft did remove VBA from Excel for Mac, but then they put it back in!



Jupyter Notebooks in Microsoft Excel. Image by the author.

Why will VBA survive? Because millions of VBA-based solutions are in use and VBA is much easier to learn and use than the alternatives.

General information on VBA programming in Excel



Important points. Knowledge of algorithms, Visual Basic programming language. Instructions are written in functions (*function*) and procedures (*sub*), which are grouped into modules. We work in VBE (Visual Basic Editor). Following is a quick-and-dirty summary of what VBA is all about:

• **Code**: You perform actions in VBA by executing VBA code. You write (or record) VBA code, which is stored in a VBA module.

• **Module:** VBA modules are stored in an Excel workbook file, but you view or edit a module by using Visual Basic Editor (VBE). A VBA module consists of procedures.

• **Procedures:** A procedure is basically a unit of computer code that performs some action.

VBA supports two types of procedures: Sub procedures and Function procedures.

• Sub: A Sub procedure consists of a series of statements and can be executed in a number of ways. Here's an example of a simple Sub procedure called Test: This procedure calculates a simple sum and then displays the result in a message box.

```
Sub Test()

sum = 1 + 1

MsgBox "The answer is " & sum

End Sub
```

• Function: A Function procedure returns a single value (or possibly an array). A Function can be called from another VBA procedure or used in a worksheet formula. Here's an example of a Function named AddTwo:

```
Function AddTwo(arg1, arg2)
AddTwo = arg1 + arg2
End Function
```

General information on programming

ALGORITHMICS - PROGRAMMING

Algorithm

- "IT" solution relating to a problem
- Sequence of actions (instructions) applied on data
- 3 main stages:
- 1. data entry (reception)
- 2. Treatments
- 3. restitution (application) of results

Program

- Transcription of an algorithm with a predefined syntax
- Visual Basic for Applications
- Same basic principles as other object languages (Java, C #, etc.)
- VBA interacts with the predefined

functions available in the Office suite

Interpreted language: + application portability; - slowness (R,VBA, Python ...)

Compiled language: + speed; - not portable

(possible solution: write once, compile anywhere; ex.Lazarus)

Pseudo-compiled language: + platform portability; - slowness (?) (principle: write once, run anywhere; ex. Java and the JIT principle)



VBA (Visual Basic for Applications) is a programming language dedicated mainly to Microsoft Office applications. It is language-based <u>Visual Basic</u>. But can only run in a Microsoft Office host application, and not on its own.

- Determine needs and set goals : what should the software do, in what framework will it be used, who will be the typical users? We write a specification with the software sponsor (<u>Note :</u> sponsor = owner; director =project manager)
- 2. Design and specifications : what are the functionalities of the software, with what interface?
- 3. Programming : modeling and coding
- 4. Tests: are we getting the expected results, the calculations are correct, is there a crash and under what circumstances? (unit tests, integration tests, etc.)
- 5. Deployment : install it at the customer's premises (verification of configurations, installation of the executable and additional files, etc.)
- 6. Maintenance: corrective, track down bugs and correct them (patches); scalable (add new features to the software: either on ergonomics, or by adding new procedures)

THE VBE EDITOR

	5 ¢	÷							Classeur1 -	Excel							• -	- 0 ×
Fichier	Accue	eil Inser	tion Mis	e en page	Formules	Données	Révision	Affichage	Développeur	Inquire	ACROBAT	Power Pivot	t Q Dites-r	nous ce que v	vous voulez fair	e Hervé HC	DCQUARD	우 Partager
Coller	X Ca	libri 5 I S	• 11 •	ÂĂĂ A	<	Stand	ard % 000 ț	-,0 ,00 Mise 00 →,0 condit	en forme M tionnelle *	lettre sous fo de tableau	orme Styles de	France Inséren Suppri Forma	r·∑ imer·↓	Trier et Rec	hercher et		
^o resse-papi	ers 🗟		Police	6	Alig	gnement	5	Nombre	Fa		Style		Cellul	es	Édition			^
Q36	Ŧ	: ×	~	fx														~
	А	В		С	D	E	F	G	н	1		J	к	L	м	N	0	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20																		
-		Feuil1	\oplus								:	•	\					•
Prêt 🔝																I II		+ 100 %









America Sector S								
Eichier Edition Affichage Insertion	Forma <u>t D</u> ébogage E <u>x</u> écution <u>O</u> utils <u>C</u> ompléments Fe <u>n</u> être <u>?</u>							
: M 🔄 🖌 🔒 🖧 🗠 🖓 (*)	▶ II 🖬 🔟 💐 🖀 🦉 🎌 🔘							
Projet - VBAProject	(Général)							
	Sub Macrol()							
 EuroTool (EUROTOOL.XLAM) VBAProject (Classeur1) Microsoft Excel Objets Evuil1 (Feuil1) Feuil2 (Feuil2) Feuil3 (Feuil3) ThisWorbook Modules Module1 	<pre>Sub Macrol() ' Macrol Macro ' With Selection.Font .Color = -16776961 .TintAndShade = 0 End With Selection.Font.Size = 12 Selection.Font.Italic = True Selection.Font.Bold = True End Sub Sub Macro2()</pre>							
Propriétés - Module1 X	' Macro2 Macro							
Alphabétique Par catégorie (Name) Module1	, End Sub							



End of Part 1



To be continued with the object model in VBA...

Thank you

Herve Hocquard(hocquard@labri.fr)

http://www.labri.fr/perso/hocquard/Teaching.html

