Convert a Macro to VBA

While macros can be created using the Macro Builder, that tool can be somewhat limited if you need to construct a more complex macro. In those situations, Visual Basic for Applications (VBA) is a more suitable platform that provides plenty of flexibility, although it can be somewhat complex for a beginner. If you would like to work with a macro using the VBA environment, you need to learn how to convert a macro created in Access to the VBA format. Over the course of this topic, you will learn how to do this.

Topic Objectives

In this topic, you will learn:

About macros and VBA

MACROS AND VBA

Access 2016 lets you convert a macro to **VBA (Visual Basic for Applications)**. This can be a useful feature if you are a more experienced developer who prefers to work with VBA code. The feature can also be of use if you want to use VBA to add additional functionality that may be hard to achieve otherwise. Finally, if you are curious about VBA, you can learn a few things by observing the resulting code after a macro has been converted. To convert a macro to VBA, open it in Design View. Then, click Macro Tools – Design \rightarrow Convert Macros to Visual Basic:



Click Convert in the resulting dialog:

Convert macro: mcrInventoryNewRcrd	×
Add error handling to generated functions	Convert
	Cancel

Then, click OK to confirm the conversion process. Your macro will now be displayed in Visual Basic for Applications. It will also be available in the Navigation pane:

All	Access Objects	~<					
Search							
Tabl	Tables						
	tblEmployees						
	tblinventory						
	tblWarehouses						
Forn	ns	~					
-8	frmInventory						
Mac	Macros						
2	mcrInventoryNewRcrd						
Modules							
2	Converted Macro- mcrinventoryN	le					

Αстіνіту

Converting a Macro to VBA

To get an in-depth look at how it works, you would like to view an existing macro in VBA.

1. To begin, open Activity 3-5 from your Exercise Files folder:

🗟 🤼 🖓 🖬 🖬	* Activity 3-5 : Database	- It\Access Databases\Activity 3-5.acc				×
File Home Create	External Data Database Tools 🛛 🖓 Tell					
File Home Cetale March Scale Scale View Partie Format Painton View Colpoard Scale All Accesss Scale Scale Tables R Scale ththranhourstory Scale Scale Immediated R Scale Immediated Scale Scale Immediated Scale Scale Immediated Scale Scale Immediated Scale Scale	External Data Database Tools Q Tell A According Y Selection Telete A Decenting Advanced a Set & Filter A Remove Set Y Toggle Filte Set & Filter	me what you want to do	P Hind → Go To* ↓ Select Find	B I U A - ♥ - A - = = = 2 Z - = Tet formating	idot - 74 100 	n Smith

If prompted, click Enable Content on the Message Bar.

2. Open mcrInventoryNewRcrd in Design View by right-clicking its listing within the Navigation pane and clicking Design View:

All	Access Obj	ect	ts 🗟	~		
Search	L			ρ		
Tabl	es			^		
	tblEmployees					
	tblinventory					
	tblWarehouses					
Forn	ns			*		
-8	frmInventory					
Mac	ros			^		
2	mcrInventoryNewRcr	4	Run			
		•	Desiden	View		
			Desid	VIEW		
			Export			•
			Rena <u>m</u>	e		
			<u>H</u> ide in	this (Group	
			De <u>l</u> ete			
		Ж	Cu <u>t</u>			
		Ē	<u>C</u> opy			
		Ĉ	<u>P</u> aste			
		*	O <u>b</u> ject	Prope	erties	

3. With the selected macro now open in Design View, you can begin converting it to VBA. Click Macro Tools – Design → Convert Macros to Visual Basic:

								Macro Too	ols Activ	vity 3-5 : Da
File	Home	e Cr	eate	External	Data	Database	Tools	Design	Q ⁻	Tell me wha
!	Single Single	Step t Macro	s tę Vi	sual Basic	+		+			the second secon
Run			43		Actions	Actions	Expand All	All	Action Catalog	Actions
		Tools				Collapse	e/Expand		Show	//Hide

4. The "Convert macro" dialog box will be displayed. This dialog box allows you to choose to include macro comments and error handling. For this example, leave the default settings unchanged and click Convert:



5. A dialog box will appear and notify you when the conversion is complete. Click OK:



6. The Microsoft Visual Basic for Applications window will now open. Double click the "Converted Macro-mcrInventoryNewRcrd" listing in the Project pane:



7. The newly converted macro will now be open in the working area. Review its code:



8. Close the Microsoft Visual Basic for Applications window and the macro object in the Access 2016 window. Save the current database as Activity 3-5 Complete. Close Microsoft Access 2016.