

The book was found

Visual Basic For Applications (VBA) 2013 Quick Reference Guide: Introduction (Cheat Sheet Of Instructions, Tips & Examples - Laminated)

Visual Basic® for Applications 2013 Introduction

Visual Basic for Applications (VBA) is a programming language built into Microsoft Office and other Microsoft applications. It is used to automate operations within those programs through a series of instructions, commonly referred to as macros. VBA instructions are usually entered into a code window within each application's linked VBA Editor. Only Excel and Word have a Macro recording feature to generate the code.

Showing the Developer Tab

Always display the DEVELOPER tab in the ribbon pane to access the VBA code and related security settings.

1. Choose FILE > OFFICE > DEVELOPER RIBBON.
2. Under CUSTOMIZE THE RIBBON, check DEVELOPER and CHECK OK.

Setting Macro Security

1. Choose DEVELOPER > VBA SECURITY.
2. Choose one of the MACRO SETTINGS from (SAFE) OK. Tip: Use the Enable All Macros option when working on your own code.

Create VBA Code

1. Press <ALT>F11 to show DEVELOPER > VBA EDITOR.
2. Open the Macro Project Explorer pane if necessary with VBA PROJECT EXPLORER > <CTRL>R.
 - There are 3 main parts of Code windows: Declaration, Module and User Form. Declaration, Module and User Form are used for customized programming. VBA code is typically stored in a Module window. See Project Explorer Names.
3. Double-click the Code window name in Project Explorer. If necessary, first open its containing folder by clicking the folder's +.
 - If the Code window does not appear, Right-click the Project Name and choose INSERT > MODULE.
4. Choose FILE > PERSONAL.
 - Enter a NAME (see Code Name Rules).
 - Select SUB to create a Subroutine, or FUNCTION to create a function. (SAFE) OK.
5. Enter the code that defines the SUB (see Saving Code).

Tip: Save time writing within Excel or Word by recording Macro actions being assigned the recorded code into its default location in your code.

Deleting a Code Window

1. Right-click its name in Project Explorer and select REMOVE WINDOW.
2. (SAFE) NO from the pop-up message unless you want to export the code for archiving.

Code Name Rules

- Code names must begin with a letter in Word and Excel but it is good practice for all Office applications.
- Subsequent characters should only be letters, numbers, or underscore characters.
- Code name and module name should have 25 characters and subroutines and functions for spaces.
- Code names that are the same as cell addresses in Excel or worksheet tab names such as A1 are not allowed.

Important Keyboard Shortcuts

Shift - Hold this key down while opening an Office application and its AutoSave code will be bypassed.

Ctrl-Break - Stops running code. This key combination is particularly useful in handling infinite loops.

Shift-F2 - Displays the VBA Editor Object Browser. Selection of an object, method or property selected in a code window.

Alt-F8 - Displays a list of macros in Excel or Word.

Microsoft and Visual Basic are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Project Explorer Names

- Inserted modules or user forms are named Module1 (1, 2, etc.) or UserForm1 (1, 2, etc.).
- **Excel:** Project name = Workbook name + Personal.xls. Code window name = Worksheet name.
- **Word:** Project name = Name of Word file name. Code window name = Subdocument.



- **PowerPoint:** Project name = PowerPoint file name. Code window name = Module1 (1, 2, etc.).
- **Access:** Project name = Database file name. Code window name = Name of form or report.

Saving Code

Code can only be saved by saving the Office application file it is stored in. Default file types for Excel, Word or PowerPoint (.xls, .docx, .pptx) cannot contain macros, therefore in those programs:

1. Choose FILE > SAVE or press <CTRL>S.
2. If a "macro-free" warning message pops up (SAFE) NO.
3. Change SAVE AS TYPE to MACRO-ENABLED (*.xlsm, *.docm, *.pptm).
4. Choose FILE, specify FILE NAME, then (SAFE) SAVE.

Saving Global Macros

When recording a macro in Excel or Word you can choose to save your work locally in the current document or globally for all documents. To save a recorded macro globally choose the following from the STORE MACRO IN drop list in the RECORD MACRO dialog box:

- In Excel choose PERSONAL MACRO WORKBOOK.
- In Word choose ALL DOCUMENTS (PERSONAL DEFAULT).

VBA code can also be saved globally if it is manually created inside the NORMAL project in Word or the PERSONAL.VLSD project in Excel.

VBA Editor Tips

- The Project Explorer pane is best used to navigate between Code windows.
- Use the Code window's Procedure drop list (see Navigating Code Window Lists) to navigate between subroutines in a long Code window.
- (SAFE) The Project Explorer folder icon () at the top of the pane to toggle its display between a simple object list and objects grouped by folders. Making this change in any one Office application will affect all Office applications.
- Close all windows and menus except the Code window while writing code to maximize its display.
- To directly edit a macro in Excel or Word press <CTRL>R, select the macro name, then (SAFE) SO F.
- Press <CTRL>O to show the VBA Editor and return to its Office application.

Password Protect Code

1. Right-click your Project name or one of its components in the Project Explorer pane.
2. Select PROJECT PROPERTIES or VBAPROJECT PROPERTIES.
3. (SAFE) PROTECTION then add your password to both the PASSWORD and CONFIRM PASSWORD boxes. This restricts code editing.
4. (SAFE) LOCK PROJECT FOR WRITING to also prevent code saving.

Note that the GENERAL tab can be used to change the Project name and description.

Understanding the VBA Language

VBA code has semantics like any language. If the structure of VBA code is not understood, the syntax of VBA will be hard to learn. The syntax of VBA can be simplified to:

Object.Method [Property]. Each such statement is a Statement, as in a single task (Method) or a change the appearance (Property) of a single object. A use in Office statements comprises a Procedure that can enter instructions in Office application (Sub) or perform an operation (Function).

Object

Each Office application object has one or more objects that can be manipulated. Excel has worksheets, workbooks and cells. Objects can be accessed in the application editor as well as in single statements.

Method

Methods are operations that can be done on an Object. In Word a Selection (Object) of words can be copied (Method).

Properties

Properties describe an Object. In Access a Label (Object) has a font color (Property) to Word a Paragraph (Object) has an indentation (Method) to a left indent value (Property).

Events

Each Object has associated Events. Events happen to an Object and are used as triggers to initiate code execution. When an event that (Object) is a Selected (Event) in a Word Document (Object) is Closed (Event) message can be programmed to pop up.

- Tip: Press <CTRL>F to view the VBA Editor Object Browser's list of Objects, Methods, Properties and Events. Each Object also has the Show > Left column displayed on Methods, Properties and Events in the work right window column. Press <CTRL>F to show the Object Browser and always use Code window.

Using Code Window Lists

Object List	Procedure List
Subroutine	Function

The Object list and the Procedure list are at the top of any non-Module code window (e.g., any Excel work sheet, Word document) from the Code window. These can be used to create code creation:

1. Select an appropriate item from the Object list.
2. Then select an appropriate item from the Procedure drop list.
3. View code into the newly created subroutines. For example, open an Access form's Code window. Select the name of a control, e.g., Text1, from its Object list and below it select from its Procedure list. A new subroutines is automatically created called Sub Text1, below appropriate instance (An Integer). It triggers actions (e.g., add text when selected) just before Text1's contents are updated. Any private code added to this subroutines can utilize embedded data and event messages before they are saved.

Construct a Code Statement

1. Select an Object (cell, Work Range, Page, Select, Object) = Range, Method = IsEmpty.
2. Select from: IsEmpty = <ISEMPTY>, &, <IS> Object = Selection, Property = Font, Color.
3. If Word delete a single character to the right with Selection.Delete Unit:=wdCharacters, Count:=1, Object=Selection, Method=Delete.
4. Word Tip: Selection.InsertTextAfter inserts a single character to the left.
5. Navigate to an Access form control with: Object = Me, Sub-Object = Text1.Text, Method = TextValue.
6. Access Tip: "Me" is a useful substitution for the current form's complete name.

© 2013 Microsoft Software Services, Inc. All rights reserved.



Synopsis

Four page laminated quick reference guide that will open the world of Microsoft Office automation in Word, Excel, PowerPoint and Access. You will learn how to modify macros, customize functions and create your own programming. Written with Beezix's trademark focus on clarity, accuracy, and the user's perspective, this guide will be a valuable resource for composing Visual Basic for Applications code. Geared toward the intermediate to advanced Microsoft Office 2013 user. The following topics are covered: Getting Started: Showing the Developer Tab; Setting Macro Security; Opening a Code Window; Deleting a Code Window; Saving Code; Project Explorer Names; Code Name Rules; VBA Editor tips; Password Protect Code & Understanding the VBA Language. Create VBA Subroutines: Useful Code Events; Navigating Code Window Lists: Object List Items and Procedure List Items. Organizing Code: Variables and Constants for Efficiency; Good Practices with Variables; Sophistication with VBA Functions: Useful VBA Function Categories; Decision Power with IF and SELECT & Eliminate Repeats with FOR and DO. Other Topics: Reach Out to Users; Troubleshooting & VBA Code Tips. A wide variety of examples provided for the following: Keyboard Shortcuts; Constructing a Code Statement; Simple Code; Variables; Code Decisions; Eliminating Repeats & Interacting with Users.

Book Information

Pamphlet: 4 pages

Publisher: Beezix Inc. (July 9, 2014)

Language: English

ISBN-10: 1939791332

ISBN-13: 978-1939791337

Product Dimensions: 11 x 9 x 0.2 inches

Shipping Weight: 0.3 ounces (View shipping rates and policies)

Average Customer Review: 4.2 out of 5 stars [See all reviews](#) (4 customer reviews)

Best Sellers Rank: #405,369 in Books (See Top 100 in Books) #115 in [Books > Computers & Technology > Programming > Languages & Tools > Visual Basic](#) #498 in [Books > Computers & Technology > Programming > Microsoft Programming](#) #813 in [Books > Computers & Technology > Programming > Introductory & Beginning](#)

Customer Reviews

These are handy to have for homework and learning or for a test if you have an open book type.

They typically are not deep in their subject but will give you the basic reference and more to refresh

your memory or get you started.

OK I guess if your a novice.

This was what I needed for a class

Handy for the price.

[Download to continue reading...](#)

Visual Basic for Applications (VBA) 2013 Quick Reference Guide: Introduction (Cheat Sheet of Instructions, Tips & Examples - Laminated) Microsoft Project 2013 Quick Reference Guide: Creating a Basic Project (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft SharePoint 2013 Quick Reference Guide: Introduction (Cheat Sheet of Instructions & Tips for On-Premises Environments - Laminated Card) Microsoft PowerPoint 2013 Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Excel 2013 Data Analysis with Tables Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Project 2013 Quick Reference Guide: Managing Complexity (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Project 2016 Quick Reference Guide Creating a Basic Project - Windows Version (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Project 2010 Quick Reference Guide: Creating a Basic Project (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Word 2013 Intermediate Quick Reference: Paragraphs, Tabs, Columns, Tables & Pictures (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Mac OS X El Capitan Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Guide) Mac OS X Yosemite Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Guide) Mac OS X Mavericks Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Guide) Windows 10 Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated) Mac OS X Lion Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Windows 8.1 Update Quick Reference Guide: Introduction (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated) Windows 8.1 Quick Reference Guide: Introduction (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated) Microsoft Windows XP Introduction Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft SharePoint 2010 Quick Reference Guide: Introduction (Cheat Sheet of Instructions, Tips & Shortcuts - Laminated Card) Microsoft Excel 2016 Introduction

Quick Reference Guide - Windows Version (Cheat Sheet of Instructions, Tips & Shortcuts -
Laminated Card) Windows 7 Quick Reference Guide (Cheat Sheet of Instructions, Tips & Shortcuts
- Laminated Card)

[Dmca](#)