

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 Word, Excel, Access 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 manually entered into a code window within each application's linked VBA Editor. Only Excel and Word have a Macro recording feature to automatically create code.

Showing the Developer Tab

Always display the DEVELOPER tab so that you can quickly access the Macros, VBA code and related security settings.

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

Setting Macro Security

1. Choose DEVELOPER > Macro Security.
2. Choose one of the MACRO SETTINGS, then OK. Tip: Use the Enable All Macros option when working on your own code.

Create VBA Code

1. Press ALT-F11 to show the VBA (Visual Basic) Editor.
2. Open the left-side Project Explorer pane, if necessary, with VIEW > PROJECT EXPLORER or click the icon. There are 3 main types of Code windows: Document, Module and Class. From Document windows, you can create a new document, module and class. From Module windows, you can create a new module, class and class module. From Class windows, you can create a new class, class module and class module. See Project Explorer Names.
3. Double-click the Code window name in Project Explorer. If necessary, first open a containing folder by clicking the folder's +.
4. If the Code window does not appear, Right-click the Project Name and choose INSERT > MODULE.
5. Choose FILE > PROJECT > VBA PROJECT > VBA PROJECT > INSERT > MODULE.
6. Enter a NAME (see Code Name Rules).
7. Select CTRL to create a Subroutine, or FUNCTION to create a function. (Ctrl+D).
8. Enter the code that creates this, SAVE (see Saving Code).

Tip: Save time writing code in Excel or Word by recording Macro actions then copying 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 (remove name).
2. (Ctrl+D) 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 in good practice for all Office applications.
- Subsequent characters should only be letters, numbers, or underscore characters.
- Good practice includes no more than 25 characters and subnames, underlines for spaces.
- Good names that are the same as call subnames in Excel or restricted VBA keywords such as ISAMFILE.

Important Keyboard Shortcuts

Alt+F11 - Show the VBA Editor Object Browser

Alt+F12 - Display the VBA Editor Object Browser

Alt+F13 - Display a list of macros in Excel or Word

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

Project Explorer Names

- Inserted modules or class forms are named Module1 (1, 2, etc.) or Sheet1 (1, 2, etc.).
- **Excel:** Project name is Workbook name or Personal.xlsb. Code window name is Worksheet name.
- **Word:** Project name is Normal or Word for name. Code window name is Paragraph.



- **PowerPoint:** Project name is PowerPoint file name. Code window name is Module1 (1, 2, etc.).
- **Access:** Project name is Database file name. Code window name is 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 (*.docx, *.doc, *.pptx) cannot contain macros, therefore in these programs:

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

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 FILE > MACRO > drop list to the PERSONAL MACRO WORKBOOK:

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

VBA code can also be saved globally if it is manually created under the PERSONAL project in Word or the PERSONAL VBA 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.
- (Ctrl+D) 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 panes except the Code window while writing code to maximize its display.
- To directly edit a macro in Excel or Word press (Ctrl+F10), select the macro name, then (Ctrl+S) to save. Press (Ctrl+Q) to close 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 > VBAPROJECT > PROPERTIES.
3. (Ctrl+D) the PROJECT > PROPERTIES dialog box to both the PASSWORD and CONFIRM PASSWORD boxes. This restricts code editing.
4. (Ctrl+D) the PROJECT > PROPERTIES to also prevent code viewing.
5. 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 English sentences can be simplified to: **Adjective Noun Adverb Verb** then the syntax of VBA sentences can be simplified to: **Object Method Property Each such sentence, or Statement, automates a single task (Method) or changes the appearance (Property) of a single object. A use of these statements comprises a Procedure that can either manipulate an Office application (Data entry) or perform a calculation (Function).**

Object: Each Office application includes many Objects that can be manipulated. Excel has workbooks, worksheets and cells. Objects can be assigned as the application method or as small as a single character.

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

Properties: Properties describe an Object. In Access a Label (Object) has a FontName (Property). In Word a Paragraph (Object) can be indented (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 Object (Object) is Closed (Event) a message can be programmed to pop up.

- Tip: Press (Ctrl+Q) at any time to reveal the VBA Editor Object Browser's list of Objects, Methods, Properties and Events. Each Object also has the Object's built-in class displayed in Methods, Properties and Events in the right-hand window column. Press (Ctrl+F10) to close the Object Browser and return to your Code window.

Using Code Window Lists

Object list Procedure list



The Object list and the Procedure list are at the top of any main VBA code window (e.g., any Excel work sheet, Word document or Access form code window). These can be used to initiate code execution.

1. Select an appropriate item from the Object drop list.
2. Then select an appropriate item from the Procedure drop list.
3. View code into the newly created subprocedure. 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 its Procedure list. A new subprocedure is automatically created called Sub Text1_Click (see VBA naming rules). It triggers actions (e.g., add data to a data) just before Text1's contents are updated. Any events added to this subprocedure can add data and return account entries before they are saved.

Construct a Code Statement

- Select an Object (e.g., All Workbooks (All), Select Object = Name, Method = Select.
- Change the font color to red.
- Select from the Object = Name, Method = Select, Object = Name, Property = Font, Color.
- In Word delete a single character to the right with Selection.Delete Unit:=wdCharacter, Count:=1, WordWrap:=True, Method:=Delete.
- Word Tip: Selection.Font.Color property deletes a single character to the left.
- Navigate to an Access form control with: Object = Me, Sub-Object = Text1, Method = Text1.Focus.
- Access Tip: "Me" is a useful substitution for the current form's complete name.

© 1998 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 Â Â See all reviews Â (4 customer reviews)

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