

## Introduction to REALbasic

### "Part 4: Writing BASIC Code in REALbasic"

by Wally Wang

Previously,

*"Part 1: The Roots of REALbasic"*

*"Part 2: Getting to Know REALbasic"*

*"Part 3: Designing a User Interface "*

Until you write BASIC code, your program won't do anything. In traditional programming, you simply write a list of commands. The more you wanted your program to do, the more commands you needed to write. Trying to write, let alone edit, such a long list of commands made programming more complicated than necessary.

REALbasic solves this problem by letting you write small, isolated programs called event procedures. An event procedure is linked to a single control, such as a push button, and runs only when a specific event occurs, such as when the user clicks on a control or moves the mouse over a control.

Every control can respond to a handful of different events, but most of the time, you need a control to respond only to one event. Responding to specific types of actions or events is known as event-driven programming because programs run only when certain events occur.

To write an event procedure, you need to specify:

- A control
- An event that you want the control to respond to

Event procedures typically perform a single task. The most common event for a push-button control to respond to is the Action event, which occurs when the user clicks on that push-button control.

The simplest BASIC command to type is simply Quit, stored in the Action event procedure. This tells your program, "When the user clicks on this control, run the Quit command, which makes your program quit running."

Event procedures typically do one of the following:

- Make your program do something, such as stop running
- Retrieve data from the user interface
- Display data on the user interface

To see how to create a simple event procedure that does nothing but make your program stop running, load up the REALbasic project (named MyFirstProject or whatever name you chose to save your project from last week).

Double-click on the Quit push button. This opens the code editor displaying the Action event procedure. Type Quit, and then click on the Window1 tab that displays your user interface, as

shown in Figure 7.

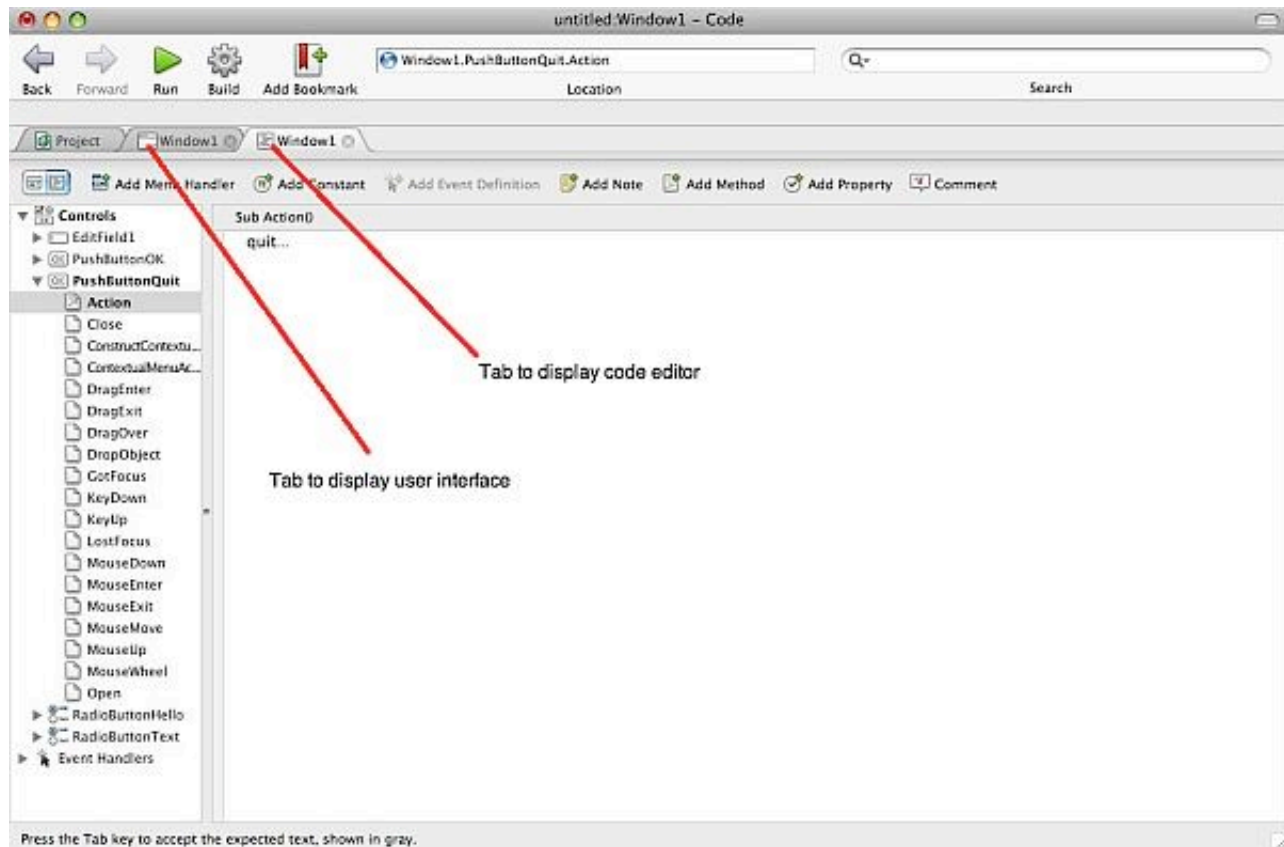


Figure 7. REALbasic's tabbed interface lets you switch between viewing your user interface and the code editor for writing BASIC code.

Double-click the OK button and type the following:

```
If RadioButtonHello.value = True then
    MsgBox "Hello, world!"
End if

If RadioButtonText.value = True then
    MsgBox EditField1.Text
End if
```

To understand the above BASIC code, here's how it works line by line.

Line 1: This line checks to see if the control, named RadioButtonHello, is selected. If the radio button is selected, then its Value property will be True. If the radio button is not selected, then its Value property will be False.

Line 2: The second line runs only if the first radio button is checked. The MsgBox command

displays the text "Hello, world!" in a message box.

Line 3: This line marks the end of the command that started on line 1 with the IF command.

Line 4: This line checks to see if the control, named RadioButtonText, is selected.

Line 5: This line runs only if the second radio button is checked. This uses the MsgBox command to display the text stored in the control named EditField1. Any text typed into the EditField1 control gets stored in the Text property of that control, so this line grabs the text in the EditField1 control and displays it in a message box.

Line 6: This line marks the end of the command that started on line 4 with the IF command.

To see how this program works, click the Project menu and choose Run. As long as you didn't make any spelling errors in typing your BASIC code, you should see your program running as shown in Figure 8.



Figure 8. Your working program.

Click the top radio button labeled Display "Hello, world!" and then click the OK button. A message box appears, displaying the phrase "Hello, world!" as shown in Figure 9.

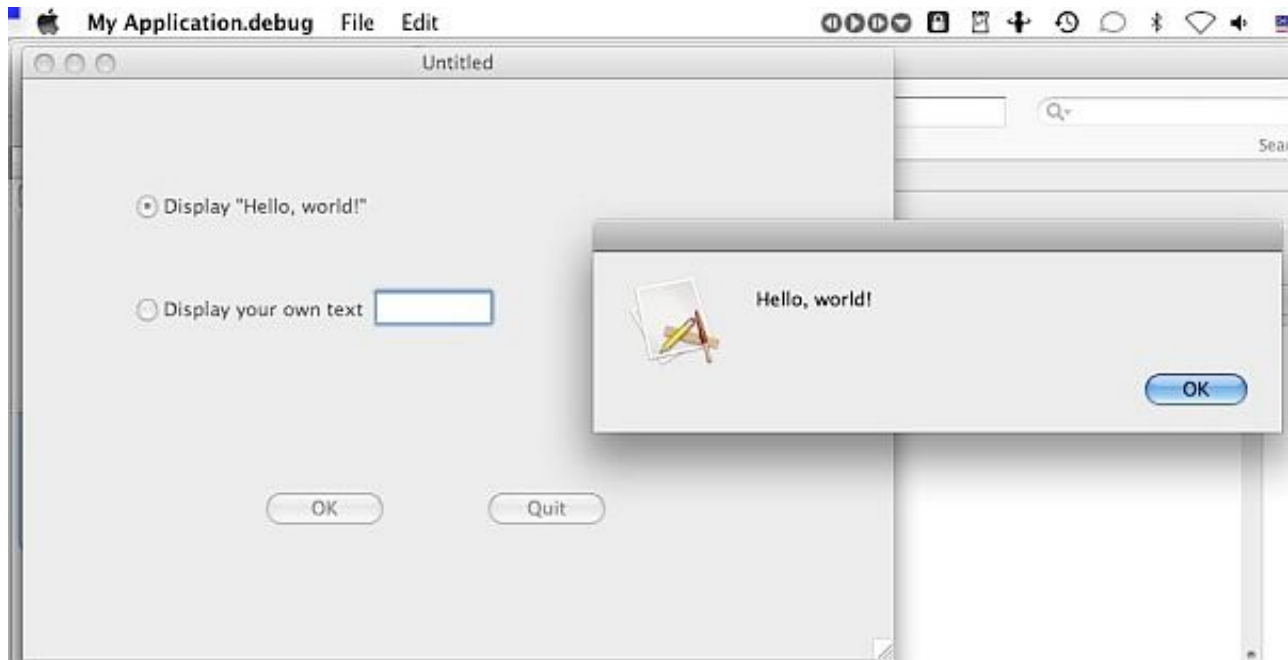


Figure 9. A message box appears, displaying the "Hello, world" text.

Click the OK button in the message box to make it go away.

Now click the bottom radio button labeled Display your own text. Click in the edit field to the right and type any text that you want, such as your own name. Click the OK button, and you'll see the text from the edit field displayed in the message box, as shown in Figure 10.

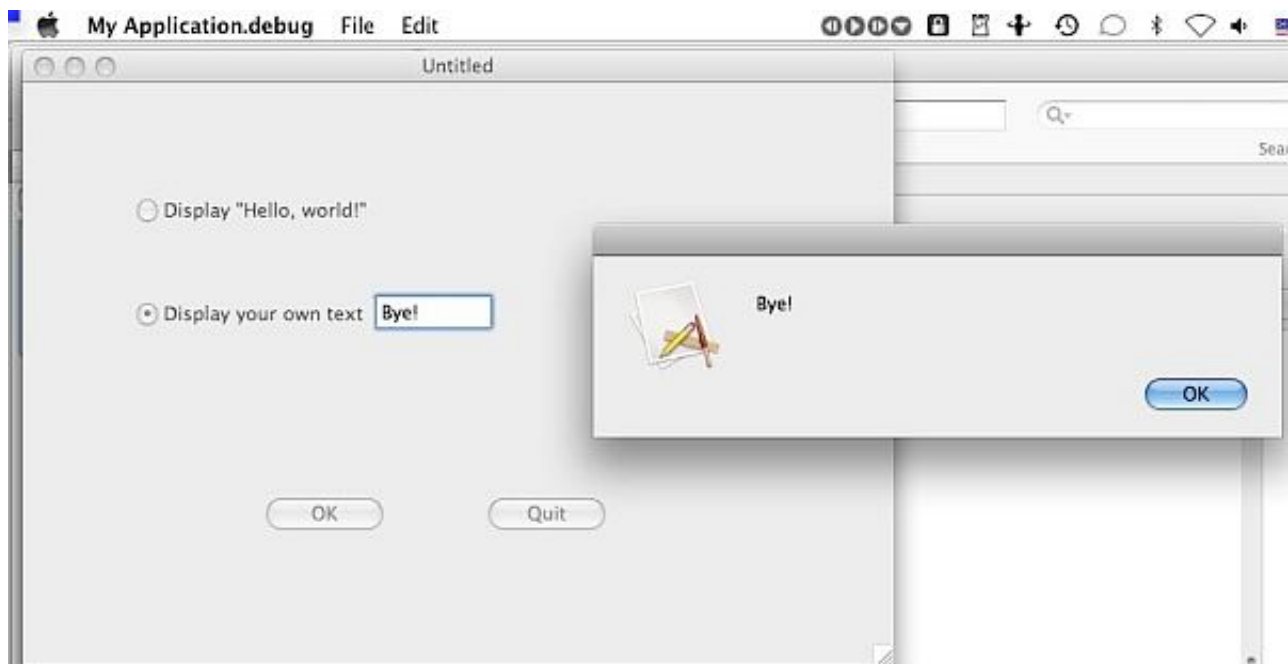


Figure 10. The text from the edit field appears in the message box.

Click the OK button in the message box to make it go away and then click the Quit button to stop

your program from running.

Congratulations! You've just created a REALbasic program by designing a user interface, customizing its properties and then writing BASIC code to make your program work.

(You can always load the Sample Program.rbp ([www.computoredge.com/images/2717/Sample%20Program.rbp](http://www.computoredge.com/images/2717/Sample%20Program.rbp)) file in case you don't want to do all the typing and customizing of the program yourself. You will need REALbasic on your computer to load and see it. Otherwise you'll just see gibberish)

Of course, there's a lot more to know before you can write more sophisticated programs in REALbasic, but you've already learned the fundamentals just by following these steps of designing a user interface, customizing its properties and then writing BASIC code to make it work.

If you've ever been interested in writing your own programs, but felt that computer programming was too difficult, you can see how REALbasic makes writing programs fast and easy. From here, start experimenting with using different controls on a user interface, dig into the REALbasic language so you can find out how to use more powerful BASIC commands, and have fun. Programming doesn't have to be complicated, and REALbasic can help turn anyone with a good idea into a real programmer in no time.

---

*In the early days, before Wally became an Internationally renowned comedian, computer book writer, and generally cool guy, Wally Wang used to hang around The Byte Buyer dangling participles with Jack Dunning and go to the gym to pump iron with Dan Gookin.*

*Wally is responsible for Microsoft Office 2007 for Dummies, Breaking Into Acting for Dummies, Beginning Programming All-in-One Reference for Dummies, and Mac All-in-One Reference for Dummies from [www.dummies.com](http://www.dummies.com), as well as, Steal This Computer Book 4.0, Visual Basic Express 2005: Now Playing, and My New Mac from [www.nostarch.com](http://www.nostarch.com). He is also the co-author of Strategic Entrepreneurism from [www.selectbooks.com](http://www.selectbooks.com).*

*Every Saturday morning from 9:00 am - 10:00 am in San Diego, you can hear Wally with fellow co-hosts Dane Henderson and Candace Lee, on the radio show CyberSports Today ([www.cybersportstoday.com](http://www.cybersportstoday.com)), which covers the video gaming industry on ESPN Radio 800 AM. Wally covers the military history side of the video game industry.*

*When not performing stand-up comedy or writing computer books, he likes to paper trade stocks with the video game Stock Reflex ([www.plimus.com/jsp/download\\_trial.jsp?contractId=1722712&referrer=wwang](http://www.plimus.com/jsp/download_trial.jsp?contractId=1722712&referrer=wwang)).*

*Wally can be reached at [wally@computoredge.com](mailto:wally@computoredge.com).*

Send mail to [ceeditor@computoredge.com](mailto:ceeditor@computoredge.com) with questions about editorial content.  
Send mail to [cwebmaster@computoredge.com](mailto:cwebmaster@computoredge.com) with questions or comments about this Web site.  
Copyright © 1997-2009 The Byte Buyer, Inc.

ComputerEdge Magazine, P.O. Box 83086, San Diego, CA 92138. (858) 573-0315

COMPUTOREDGE<sup>®</sup> ONLINE

[www.computoredge.com](http://www.computoredge.com)

04/24/2009