

A C Programmer's Guide to Getting a little work done in Visual Basic 6.0

Jennifer Kay, Rowan University Computer Science Department

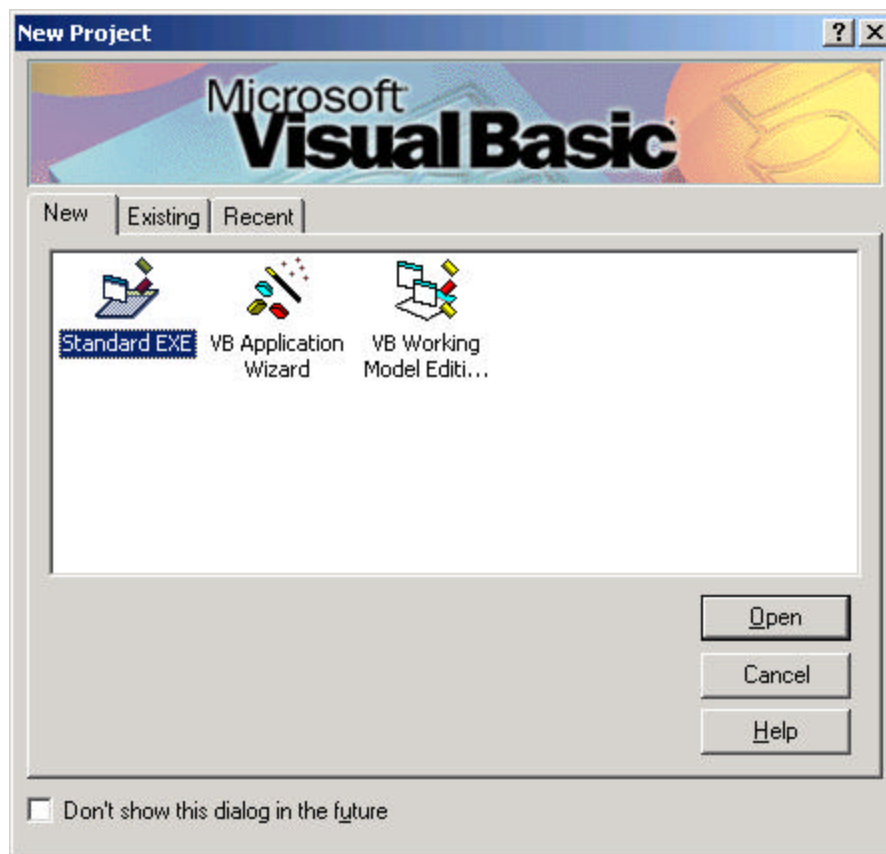
Introduction

The goal of this document is to allow someone with a little knowledge of C write very simple programs in Visual Basic. By no means does it tell you all you need to know to be a good Visual Basic programmer, but I hope it gives you a feel for the language. After reading this, if you want to write programs with lots of bells and whistles, you can buy a Visual Basic book and teach yourself the fancy stuff.

Making a new Project

When you start up Visual Basic it should bring up the new project window (see below). (If it doesn't display the window when you start it up, select "New Project" from the file menu.)

- Click on Standard EXE and say OPEN



Copyright (c) 2002 Jennifer Kay. Permission is granted to make copies of this for non-profit educational purposes as long as it is distributed for free (or for a minimal copying fee), this copyright notice remains intact, and an email is sent to kay@rowan.edu summarizing your use of the document.

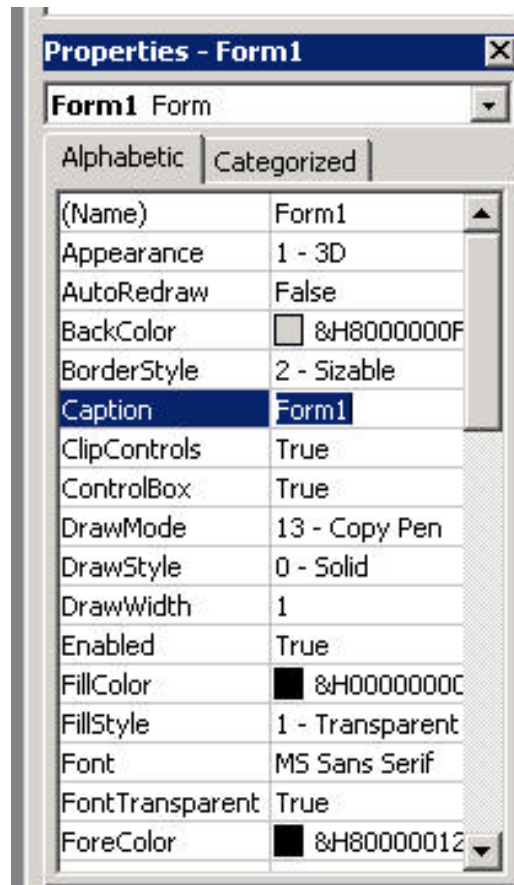
Creating a window

In Visual Basic, before you write any code, you have to design your form. Unlike in C, you can't do anything in Visual Basic without a form. You can design fancy forms, but we're just going to do a very simple one that will allow you to get input from the user and display output.

When your new project opens, you get a blank form labeled Form 1. This form is actually the window that your program will run in. You can see what it looks like running by pressing the "play" button (looks like a triangle) at the top of the page. Pressing the "stop" button (the square) stops the program.

Most people like their programs to have a special name at the top, instead of just form 1.

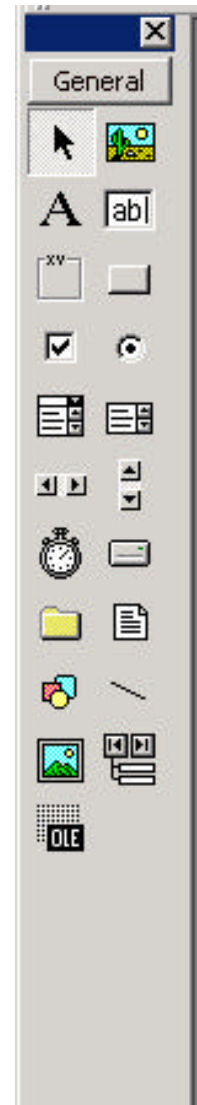
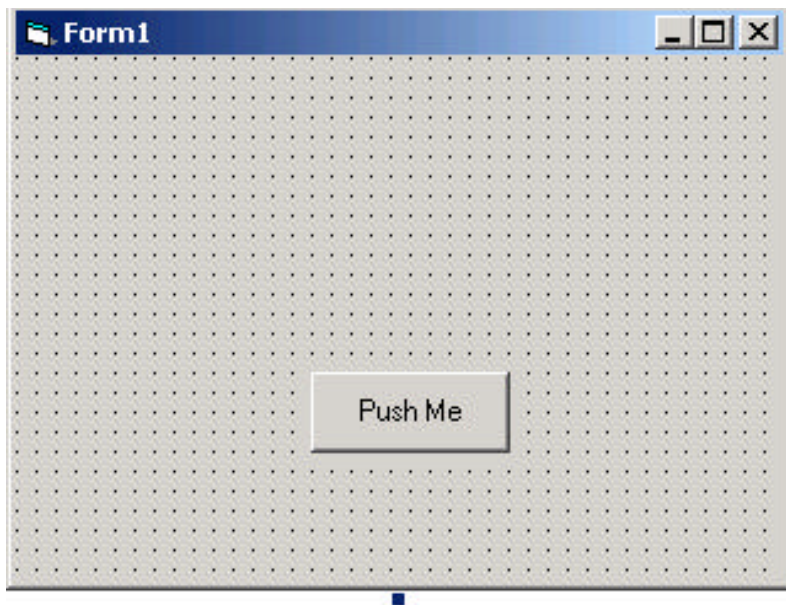
- Go to the properties menu on the right side of the Visual basic window. It looks like the picture below.
- Highlight the words "Form1" under caption and change it to "My Way Cool Program"
- Hit play to look at your cool form, and then hit stop.



Adding a Button

Let's give your user something to click

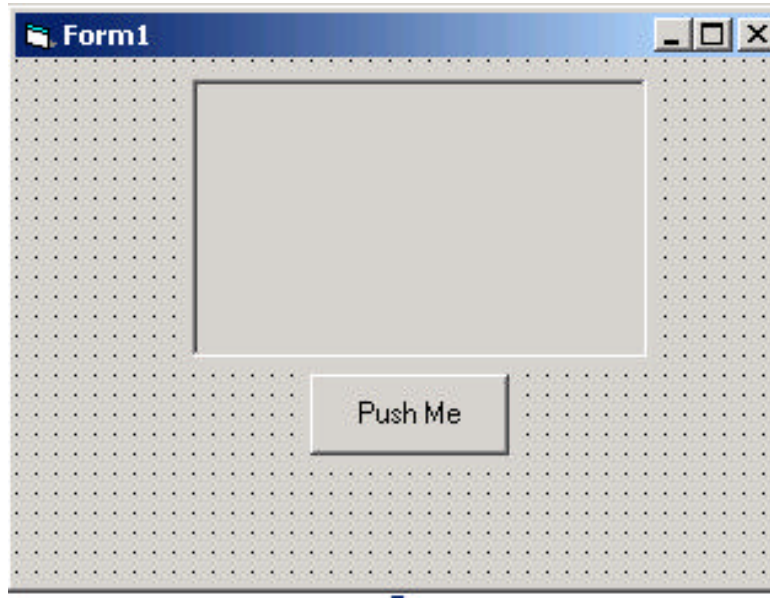
- On the toolbox (picture at right) double click on the button (right below the box labelled "ab")
- Drag the button down a bit on your form so it's on the lower third of the form.
- Go to the properties box. Change the **Caption** of your button to **Push Me**
- Also in the properties box, change the **(name)** of your button from Command1 to **my_button**
- Your form should now look like the picture below.
- Run your program. You will now have a button you can click (but that doesn't do anything!)



Writing “Hello, World” (or, Adding a picture box)

Now let’s start doing a little programming. We’re going to write a program that prints “Hello, World” on the screen when you press the button.

- On the toolbox (picture at right) double click on the desert picture to get a picture box.
- Slide it down and stretch it out a bit so that your picture looks like the one below



- Go to the properties box. Change the name of your picture box (at the top, under (Name)) from Picture1 to
my_output

Here’s how we want your program to work: When the user pushes the button, we want to print something on the picture window. Do this as follows:

- Double click on your button (that says “push me”) to tell visual basic you want to program what happens when the user clicks the button.
- Write the following program:

```
Private Sub my_button_Click()  
    my_output.Print "Hello, World"  
End Sub
```

This program basically says “when my button gets clicked, print “Hello, World” on the picture box called my_output. Close the program window, and run your program.

If you want, add another line right after the first print line that says

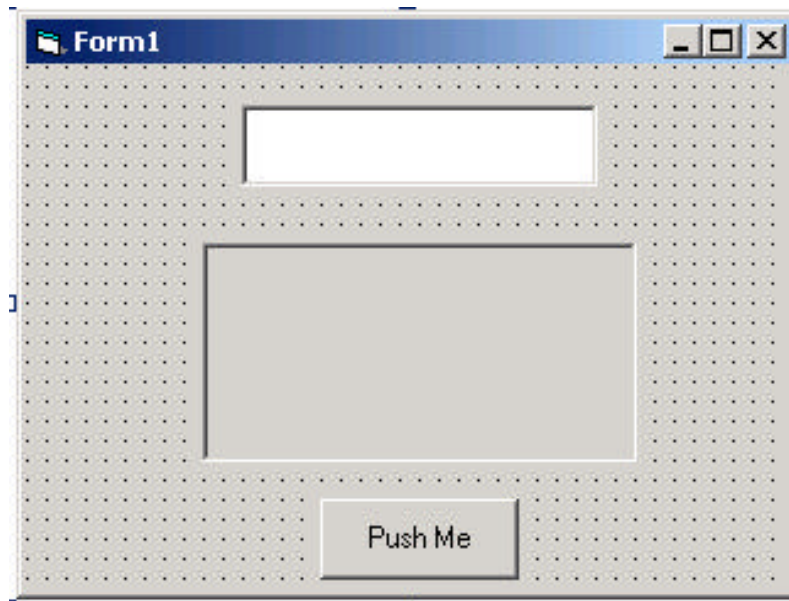
```
my_output.Print "How are you?"
```

and run your program again.

Adding a text box

It's convenient to give your user a place to put information.

- On the toolbox (picture at right), double click on the “text box” button (labelled “ab|”)
- Go to the “**Text**” property, and erase the value on the right.
- Go to the (**Name**) of your textbox, and change it to
my_input
- Move the objects around so that your form looks like the picture below.
- Run your program. Now you can type whatever you want in the text box. It doesn't do anything, but it's fun to type.



Adding A label

Labels help your user know that they are expected to enter data.

- Click once on the capital letter A and drag it on your form to add a label. (You could also double click like before)
- Change the **caption** (on properties) of your label to:

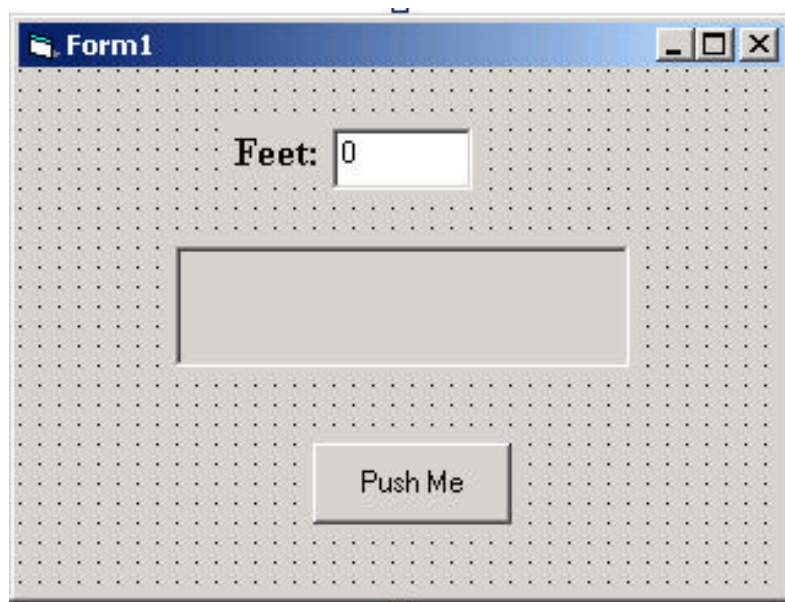
Feet:

- Click on the **font** property and then click on the 3 dots and change it to Times New Roman 12 point bold
- Use the drop down menu (triangle) on the properties to switch to the my_input textbox (or double click on the text box), and change the text to

0

(i.e. the number zero)

- Modify your form so it looks like the picture below



Some Programming Information

At this point you're almost ready to write some more complex visual Basic Programs. Here's some basic information you need to know. Here are some common variable declarations in C and the corresponding declarations in VB. Notice that VB does not use semicolons in the declarations.

Declaring Variables

Declaration in C	Declaration in Visual Basic
<code>int my_favorite_num;</code>	<code>Dim my_favorite_num as Integer</code>
<code>float weight;</code>	<code>Dim weight as Single</code>
<code>float height;</code>	<code>Dim height as Single</code>

Assigning values to variables in VB is similar to C. Note, there are no semicolons in VB assignment statements.

Some Assignment Statements

Assignment in C	Assignment in Visual Basic
<code>my_favorite_num = 5;</code>	<code>my_favorite_num = 5</code>
<code>weight = 6.2;</code>	<code>weight = 6.2</code>
<code>height = weight;</code>	<code>height = weight</code>
<code>weight = weight + 2;</code>	<code>weight = weight + 2</code>

Getting input from the user is a little complicated in VB, because you have to know where they are putting it. On our form, the user will be typing in the my_input window. Here's how you do it.

Getting input from the user

Input in C from stdin	Input in the my_input text box in VB
<code>scanf("%d", &my_favorite_num);</code>	<code>my_favorite_num = Val(my_input.Text)</code>
<code>scanf("%f", &height)</code>	<code>height = Val(my_input.Text)</code>

NOTE: If you want to input more than one thing from the user, you need to create more input boxes.

Comments

Comments in C	Comments in VB
<code>/* This is a comment in C */</code>	<code>'Comments in VB Start with a quote 'and go to the end of the line. 'Multi-line comments need a quote 'at the start of each line</code>

A more complex program: converting feet to inches

Double click on the button again to bring up the code window. Edit the code so it looks like this:

```
Private Sub my_button_Click()  
  
    'Step 1: Declare two variables  
    Dim how_many_feet As Single  
    Dim computed_inches As Single  
  
    'Step 2: Get your input  
    how_many_feet = Val(my_input.Text)  
  
    'Step 3: Convert from feet to inches  
    computed_inches = how_many_feet * 12  
  
    'Step 4: Clear the picture window  
    my_output.Cls  
  
    'Step 5: Print your answer in the picture window  
    ' Note that semicolons allow you to print things on  
    ' the same line (i.e. they stop a \n)  
    my_output.Print how_many_feet;  
    my_output.Print " feet is the same as ";  
    my_output.Print computed_inches;  
    my_output.Print " inches!"  
  
End Sub
```

More Programming Information

Conditionals in C and VB

Conditionals in C	Conditionals in VB
<pre>if (expression) { code; more code; even more code; }</pre>	<pre>If expression Then code more code even more code End If</pre>
<pre>if (expression) { code; more code; } else { some more code; and a bit more; }</pre>	<pre>If expression Then code more code Else some more code and a bit more End If</pre>

Comparison Operators

C	VB
>	>
>=	>=
<	<
<=	<=
!=	<>
= =	=

Another program to try:

```
Private Sub my_button_Click()  
  
    'Step 1: Declare two variables  
    Dim how_many_feet As Single  
    Dim computed_inches As Single  
  
    'Step 2: Get your input  
    how_many_feet = Val(my_input.Text)  
  
    'Step 3: Clear the output window  
    my_output.Cls  
  
    'Step 4: If the input is negative, complain  
    '         Otherwise do the computation  
  
    If how_many_feet < 0 Then  
        my_output.Print "Please enter a positive number for feet!!!"  
    Else  
        computed_inches = how_many_feet * 12  
        my_output.Print how_many_feet;  
        my_output.Print " feet is the same as ";  
        my_output.Print computed_inches;  
        my_output.Print " inches!"  
    End If  
  
End Sub
```

Loops

Loops in C and VB

C	VB
<pre>while (expression) { code; more code; }</pre>	<pre>Do While code more code Loop</pre>
<pre>for (i=5; i<=23; i++) { code; more code; }</pre>	<pre>For i = 5 to 23 code more code Next i</pre>
<pre>for (i=10; i<=30; i=i+5) { code; more code; }</pre>	<pre>For i = 10 to 30 Step 5 code more code Next i</pre>

Acknowledgements

Thanks to Schneider's "An Introduction to Programming Using Visual Basic 6.0" for some introductory ideas.