

What Is Visual Basic and Why do I need it?

Visual Basic is Easy to learn Programming language.
With Visual Basic you can develop Windows based applications and games.
Visual Basic is much easier to learn than other language (like Visual C++),
and yet it's powerful programming language.

Visual Basic suits more for application developing than for Games developing.
You can create sophisticated games using Visual Basic, But
If you want to make a really advanced professional game like Quake 2,
You may choose other language (like C++), that would be much more
harder to program with.
However, Visual Basic will be probably powerful enough to suit all your application
and games programming needs.

The advantages of Visual Basic:

- 1) It's simple language. Things that may be difficult to program with other language can be done in Visual Basic very easily.
- 2) Because Visual Basic is so popular, There are many good resources (Books, Web sites, News groups and more) that can help you learn the language. You can find the answers to your programming problems much more easily than other programming languages.
- 3) You can find many tools (Sharewares and Freewares) on the internet that will Spare you some programming time.
For example, if you want to ping a user over the internet in your program,
Instead of writing the ping function yourself, you can download a control
that does it, and use it in your program.
Compare to other languages, Visual Basic have the widest variety of tools
that you can download on the internet and use in your programs.

The disadvantages of Visual Basic:

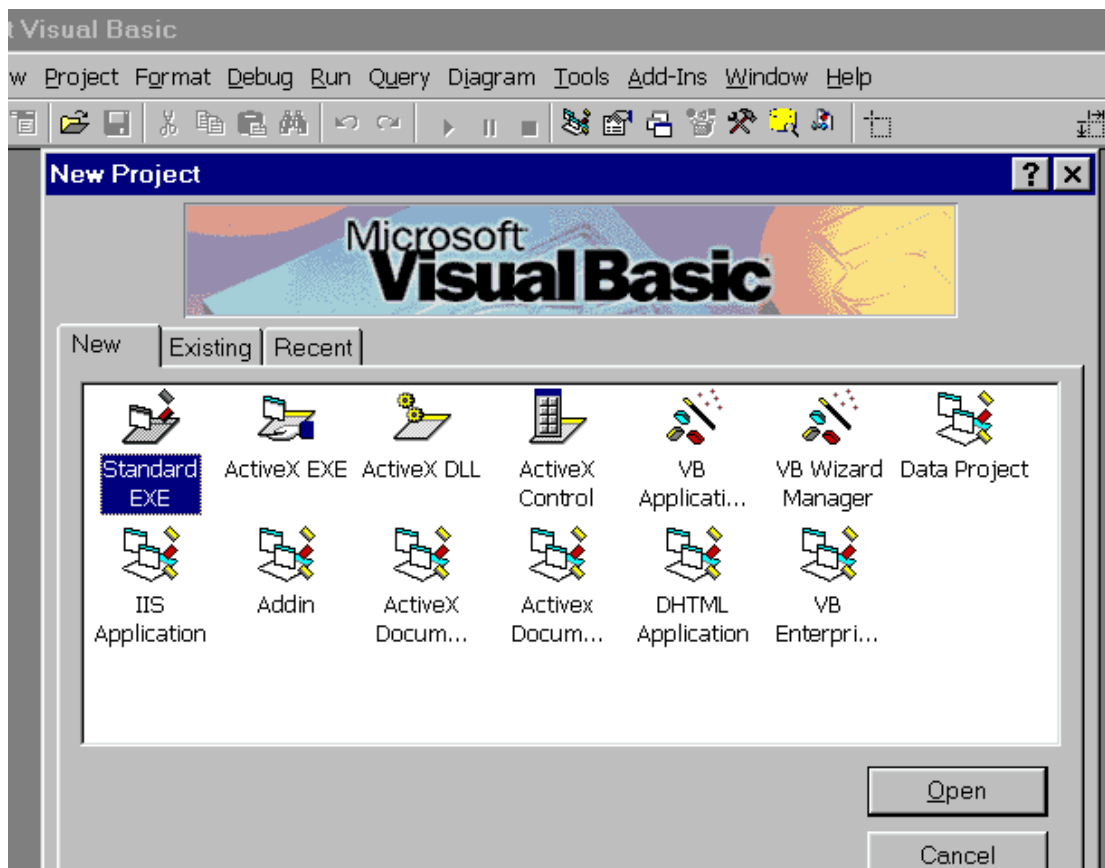
- 1) Visual Basic is powerful language, but it's not suit for programming really sophisticated games.
- 2) It's much slower than other languages.

Getting Started

Note that all the images in this tutorial taken from Visual Basic version 6.0. If you using other version of Visual Basic, the images that you will see may be a little different.

Run the Visual Basic program. The first thing you see is:

Figure 1

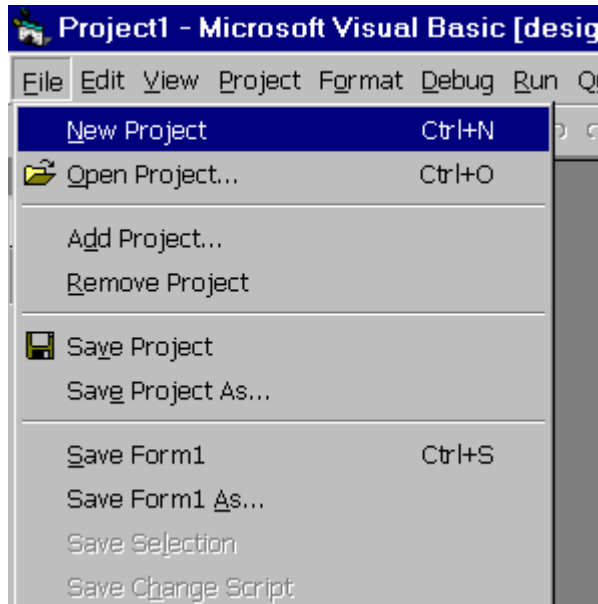


Here you choose what is the kind of the program you want to create. For creating a simple Visual Basic program, choose the Standard EXE, and press the Open Button.

(If Figure 1 is not the first thing you see when you run Visual Basic, choose from the Visual Basic menu File->New Project (Figure 2))

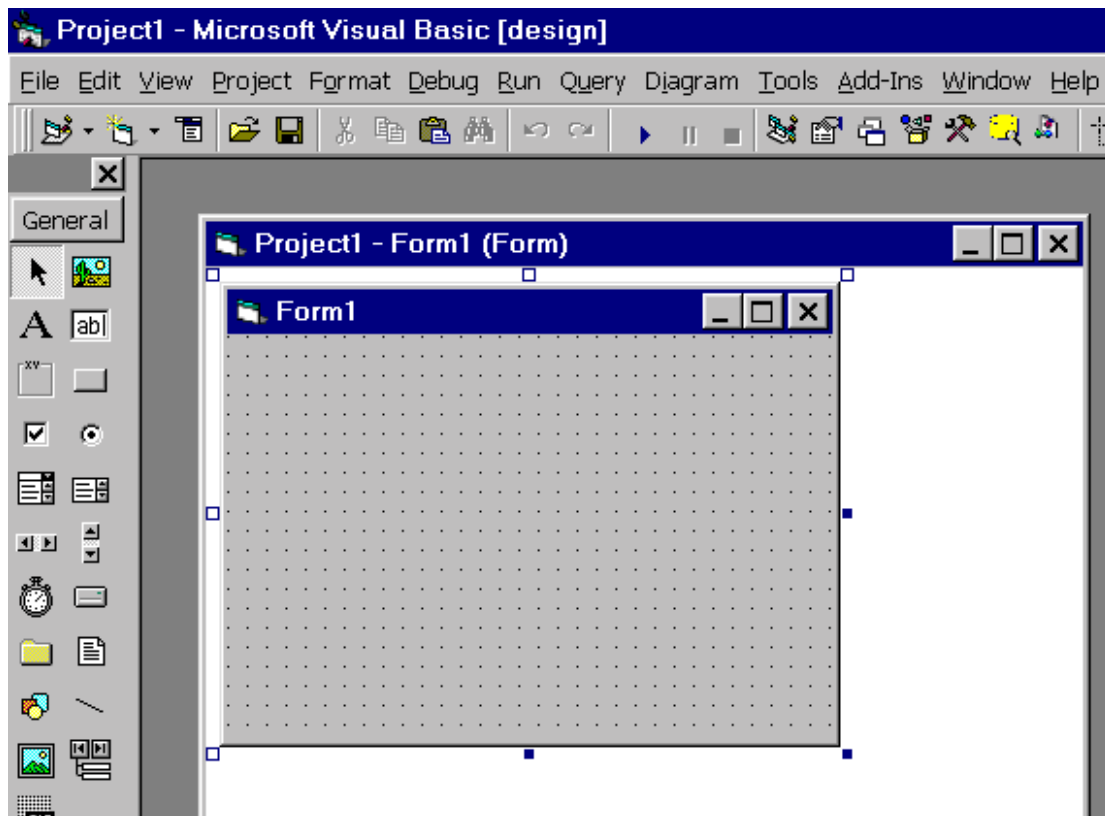
CTP108 - Programming for Business
Lecture Notes
Week1

Figure 2



After you've clicked the Open button, you will see:

Figure 3

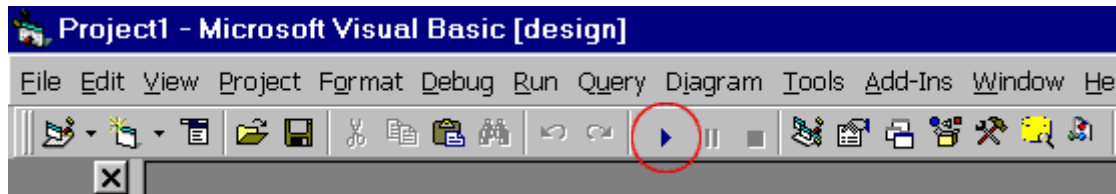


Getting Started (Continue)

Look at the form with the title bar Form1.
This is how your program will look like.
Everything you will place on this form will appear in your program.
As you can see, your form is currently empty.
You didn't program anything yet, but lets run the program!

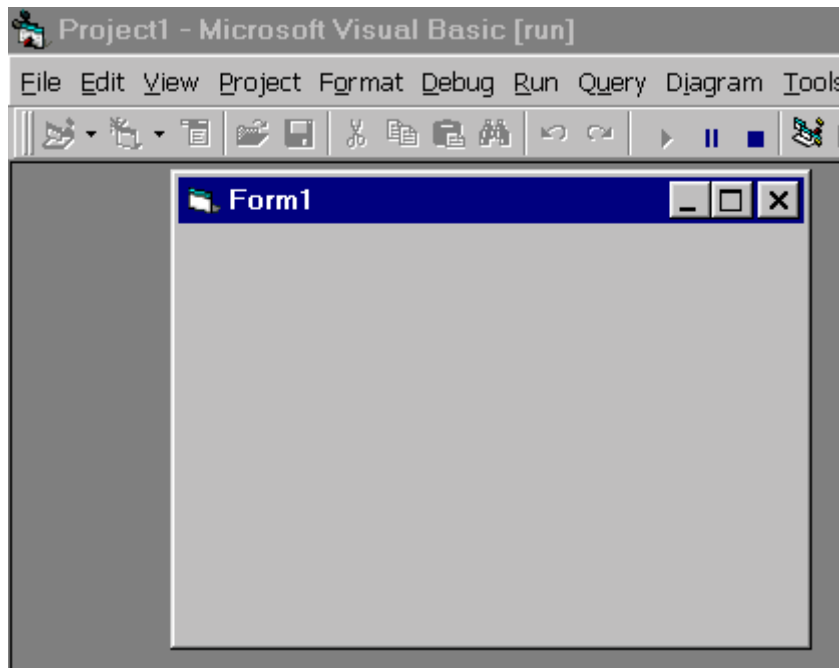
Click on the Play button in the toolbar (Figure 4)

Figure 4



Now your program is running (Figure 5):

Figure 5



As you can see, the form is empty.
You can move the form, minimize and maximize it.

To stop the program from running, you can click on the Stop button in the Toolbar (Figure 6), or click the form's X button (Figure 7).
It's very recommended that you will always close your programs using the form's X button instead of the Visual Basic Stop button.
It's like shutting Windows from the Start button, instead of Pressing the Power button.

Figure 6

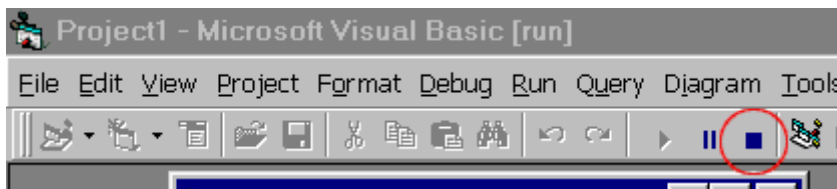
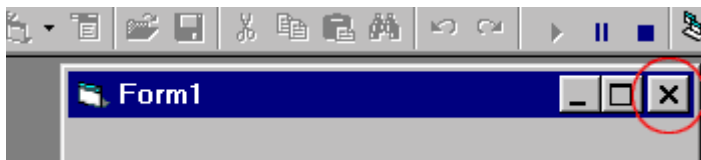


Figure 7

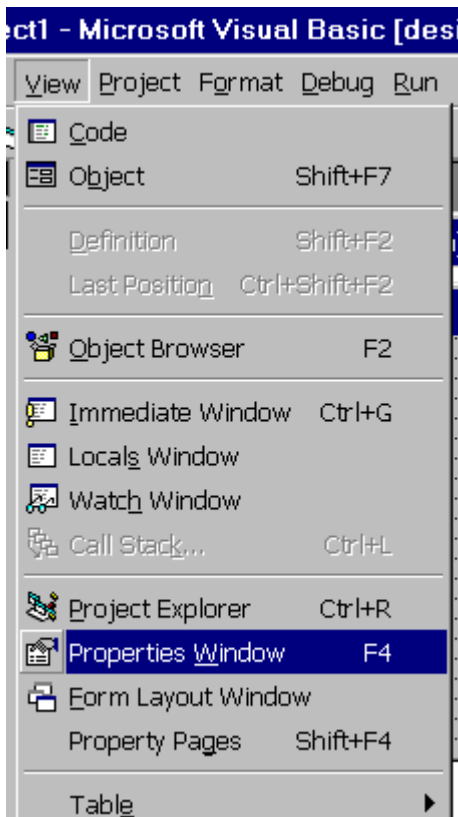


Learning about Properties

Every component (form is component for example) has properties, that determine its look and its functioning.
Properties can be the component color, width, height and more.

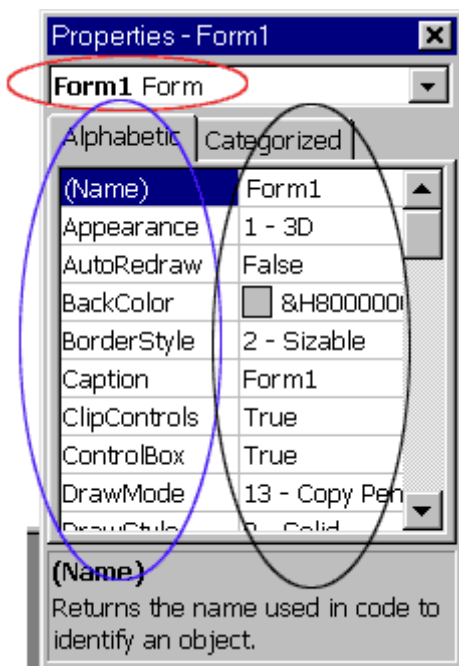
To see the form properties, select from the menu View->Properties Window (Figure 8).

Figure 8



The properties window looks like this:

Figure 9



CTP108 - Programming for Business
Lecture Notes
Week1

In the red circle you can see the component name:
These are the properties of a Form, and the form's name is Form1.

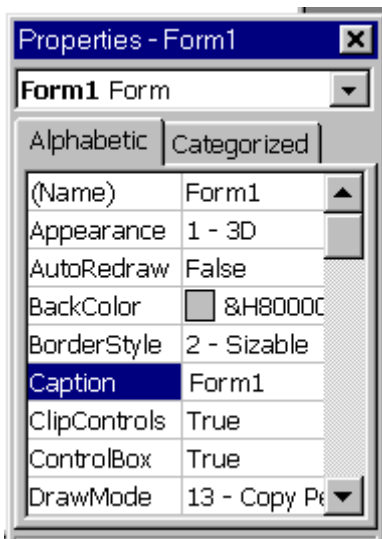
The column marked by the blue circle contains the form's property names:
The form has Name property, Appearance property AutoRedraw property, and more.

The column marked by the black circle contains the form's property values:
The form's Name is Form1, The form's Appearance property is 1 - 3D, and so on.

You can change the form's properties.
For example, lets change the form's Caption property.
What is the Caption property? The Caption is the text that appears on the
Form's title bar.
Right now the text that appears on the form's title bar is Form1 (Figure 5).

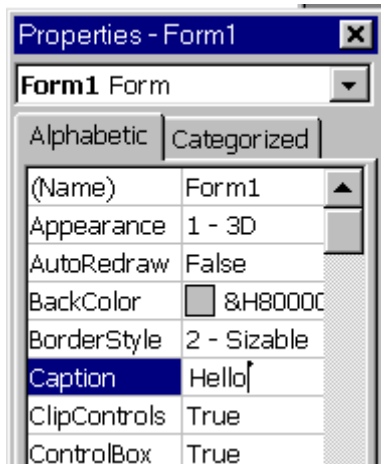
To change the Caption property, simply click on the Caption property in the
form's properties names column (Figure 10).

Figure 10



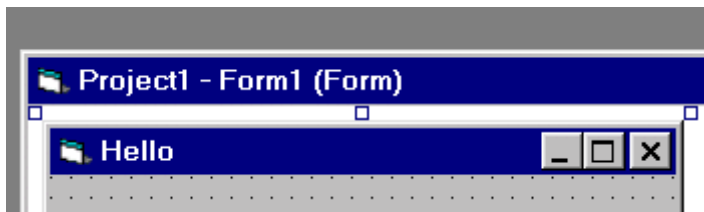
As you see, the current Caption is Form1.
Delete the "Form1" text and type instead of it "Hello" (Figure 11).

Figure 11



Now run the program using the Play button (Figure 4).
You will see that the text on the form's Title bar is Hello (Figure 12).

Figure 12



Try to change others properties:

Change the form's BackColor property to change the form's background color,
Change the Icon property to change the icon that appears on the form's title bar.
You can try and change every property, and in this way learn what does it do.

Adding Controls to the Form

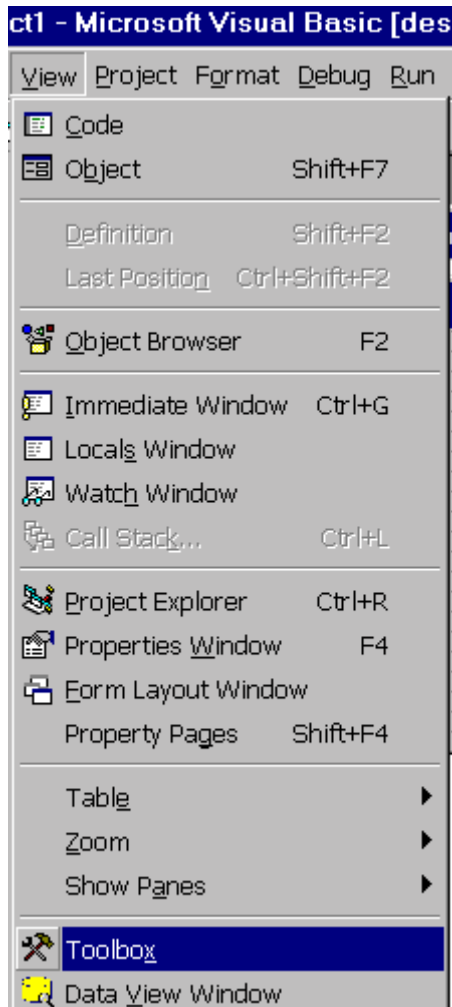
There are many controls you can use with your program:

Buttons, Text Boxes, Pictures, Scroll Bars and more.

You can see all these controls in the Toolbox.

To see the Toolbox, Select from the menu View->Toolbox (Figure 13).

Figure 13



And you will see the Toolbox (Figure 14):

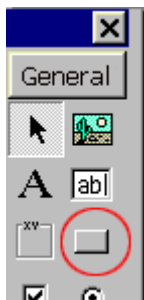
Figure 14

CTP108 - Programming for Business
Lecture Notes
Week1



Now lets add a button to your form.
Buttons in Visual Basic called "Command Buttons".
To add a button, click on the Command Button icon in the Toolbox (Figure 15).

Figure 15



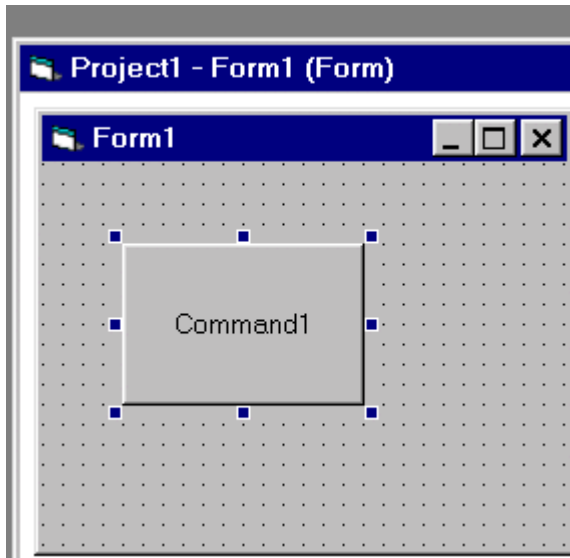
As results, the Command Button icon will look pressed (Figure 16).

Figure 16



Now click on the form with the left mouse button, and hold the button down while moving the cursor.
You will see a rectangle.
Release the mouse button, and in the place of the rectangle you will see a button (Figure 17).

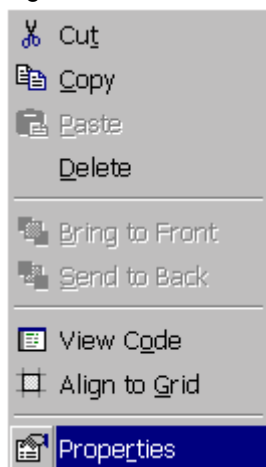
Figure 17



Changing the button's Properties

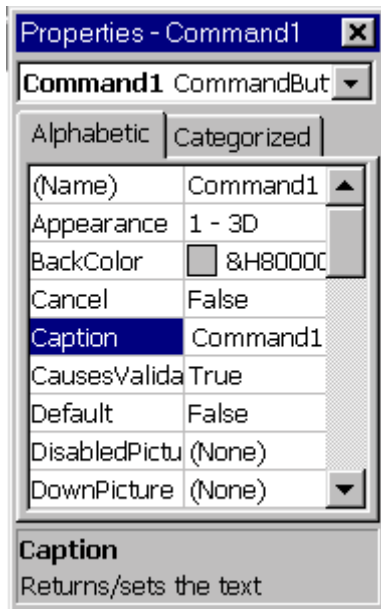
Now you have a button on your form.
You can change its location by dragging it, and change its size by dragging the Blue Hot Spots that found on each one of its corners.
Run the program by clicking the Play button.
You can see that you have a button on your form, that you can click on it, But it still doesn't do anything.
To see the Command Button's Properties window, click on it with the right mouse button and select Properties for the pop-up menu (Figure 18).

Figure 18



The Command Button's Properties window (Figure 19):

Figure 19



As you can see, In the top of the properties window, right under the title bar, appears "**Command1** CommandButton".

Command1 - because it's the name of this specific Command Button (look at the name property).

CommandButton - because this is this component type.

Note that two components can't have the same name.
If you will add another Command Button to your form, its name will be "Command2" by default.

If you want you can change the Name of any component, by simply changing the component's Name property.

Now lets change the Command Button's Caption property.
The caption property is the text that appears on the Command Button.
Change the Caption property to "Click Here", and you will see that "Click Here" appears on your Command Button.

Change the Command Button's BackColor property to blue (or other color).
The change won't take affect, untill you will set the Command Button's Style property to 1 -Graphical.

You can now play a little bit with the Command Button's properties, this is the best way to learn what every property does. You can also add other controls from the Toolbox to your form, And play around with their properties.

Learning about Events

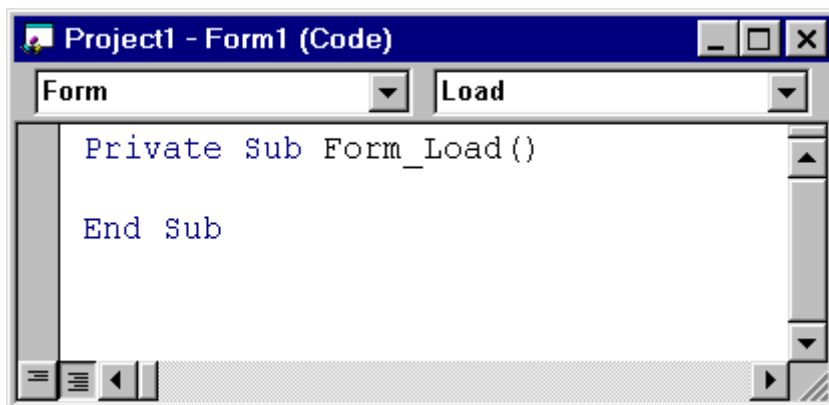
Visual Basic is "Event Driven" language.
What does it mean?

Everything that happening launches an event.
You've moved the mouse? The "MouseMove" event has been launched.
You've pressed a key? The "KeyPress" event has been launched.

You can program the events.
When the mouse moves, you can change the Form's color (for example),
and when a key is pressed, You can play a MP3 file.

To start programming the events, double click on the form.
You will see the "Code Window" (Figure 20).

Figure 20



The Code Window opened with the Form_Load event.
The Form_Load event occurs when the form is loaded, and this happening when you start the program.
So the code that you will enter to the Form_Load event will be launched when the program is being started.

The code that belongs to the Form_Load event should be placed between `Private Sub Form_Load()` and `End Sub`

The Form_Load event should look like this:

`Private Sub Form_Load()` (The beginning of the Form_Load event)
This is the code that belongs to the Form_Load event
`End Sub` (The end of the Form_Load event)

Learning about Events (Continue)

Let's program the Form_Load event.

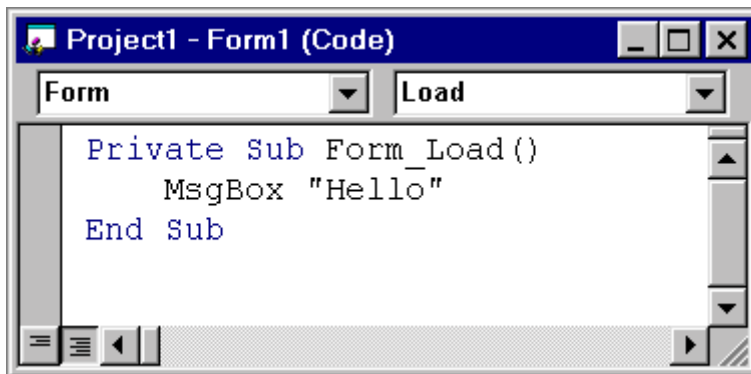
"MsgBox" is Visual Basic command that launch a message box.
for example, the line:

```
MsgBox "Hello"
```

Will launch a message box with the text "Hello".

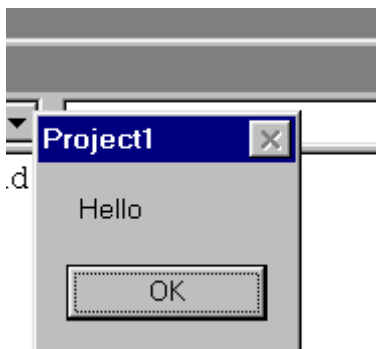
Insert the line MsgBox "Hello" to the Form_Load event (Figure 21).

Figure 21



Now run your program using the Play button.
When the program is started, a message box with the
text "Hello" is appear (Figure 22).

Figure 22

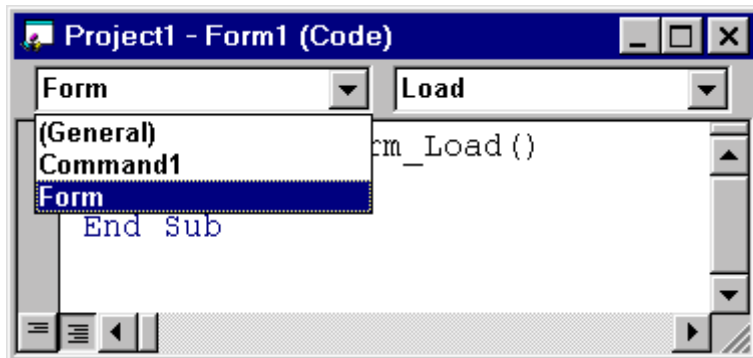


More Events

The Form has more events besides the Form_Load event.
How can you find them?

Click on the Drop-Down List that found in the upper left corner of the Code Window, where appears right now the text "Form" (Figure 23).

Figure 23

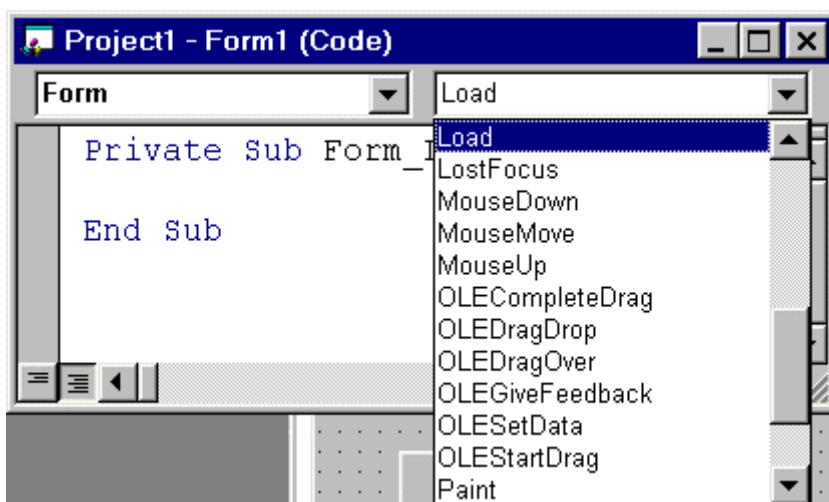


You will see a list of the components that found in your program. You have 1 command button with the name "Command1" and 1 Form. Here you select which component's event you want to program. We want to program a form's event, so select "Form" from the list (Figure 23).

Which events the form has?

Click on the Drop-Down List that found in the upper right corner of the Code Window, where appears right now the text "Load" (Figure 24).

Figure 24



You will see the complete list of the form's events: Load, LostFocus, MouseDown, MouseMove and more.

Lets program the Form_Unload event.

Select "Unload" from the form's events list.

The Form_Unload event occurs when the form is being unloaded,

and this happening when you close the program (Using the form's X button (Figure 7)).

So the code that you will write in the Form_Unload event will be launched when you close the program.

Insert the following line to the Form_Unload event:

```
MsgBox "GoodBye"
```

After you've inserted this line to your Form_Unload event, the Form_Unload event should look like this:

```
Private Sub Form_Unload(Cancel As Integer)
MsgBox "GoodBye"
End Sub
```

Run the program.

When the form is being loaded at the very start, The Form_Load event is being executed and a message box with "Hello" text is popping.

When you close the program by clicking the form's X button, the Form_Unload event is being executed and a message box with "GoodBye" text is popping.

The Command Button's Events

Now lets program some of the Command Button's events. Select "Command1" from the components list (Figure 23). Check which events the Command Button has by clicking the Events list (Figure 24).

We want to execute a code when the user is clicking on the button, So lets program the Command Button's Click event.

Select "Click" from the Events list.

Insert the following line to the Click event:

```
MsgBox "You have clicked on the button!"
```

After you inserted this line to the Click event the Click event should look like this:

```
Private Sub Command1_Click()
MsgBox "You have Clicked on the button!"
End Sub
```

As you can see, the Command Button's Click event called "Command1_Click", because the name of the specific Command Button that we program its Click event is "Command1".

If we had program the Click event of a Command Button with the name "BlahBlah7", the Click event would be called "BlahBlah7_Click".

Every component has its own unique events, therefore if you had 5 Command Buttons on your form, every one of them has its own unique Click event.

Run the program, and click the button.

When you clicking the Command Button with the

name "Command1", The Command1_Click event is being executed, and a message box with the text "You have Clicked on the button!" is popping.

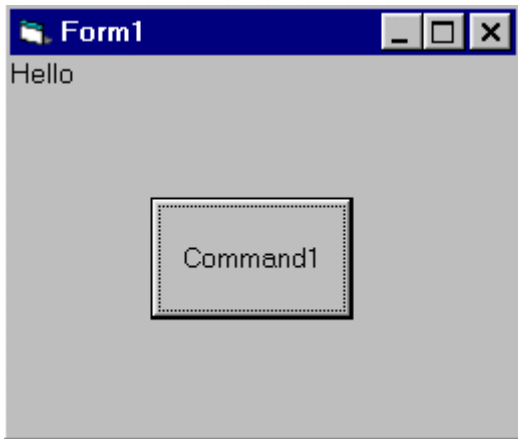
The Command Button's Events (Continue)

To learn about more events, we will use the "Print" command. The "Print" command simply writes a text on the form. For example, the following line:

```
Print "Hello"
```

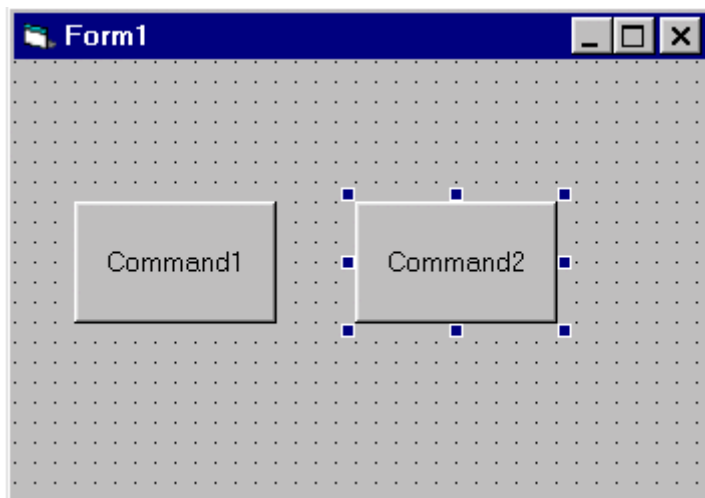
Will write "Hello" on the form (Figure 25)

Figure 25



Add another Command Button to your form. The New Command Button's name is "Command2" by default (Figure 26)

Figure 26



CTP108 - Programming for Business
Lecture Notes
Week 1

Now, rewrite the Command1 Click event and insert the following line to it:

```
Print "This is Command1"
```

Select "Command2" (This is the name of the second Command Button) from the Components list (Figure 23), And select "Click" from the Command2 events List.

Enter the following line to the Command2_Click event:

```
Print "This is Command2"
```

After you've done so, your code should look like this:

```
Private Sub Command1_Click()  
Print "This is Command1"  
End Sub  
Private Sub Command2_Click()  
Print "This is Command2"  
End Sub
```

Run the program.

When you are clicking on Command1 Button, the text "This is Command1" appears on the form, and when you are clicking on Command2 Button, the text "This is Command2" appears on the form.

.