Visual Basic Programming
(Mid-West/Great Lakes ARC/INFO Users Group Conference)

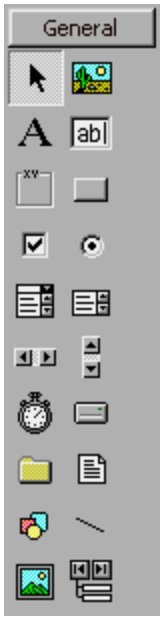
Designing Screens in Visual Basic

What is programming?

Programming means we are providing instructions to a computer to interact with it. Visual Basic is a tool that translates human-understandable instructions into machine-understandable instructions.

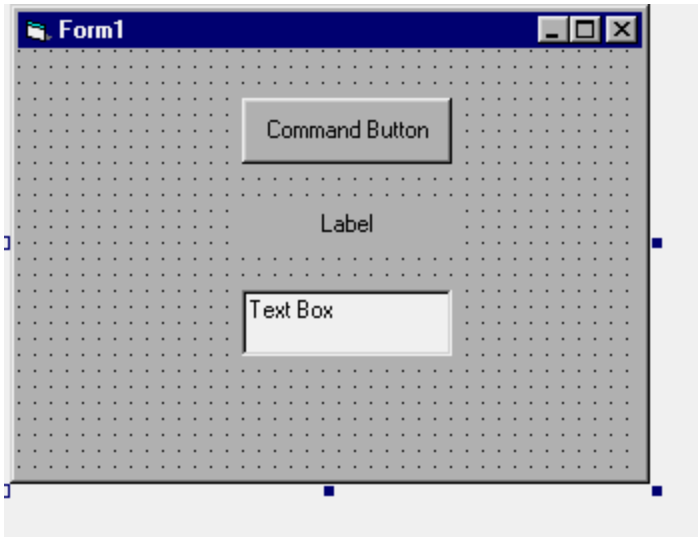
How do we design screens?

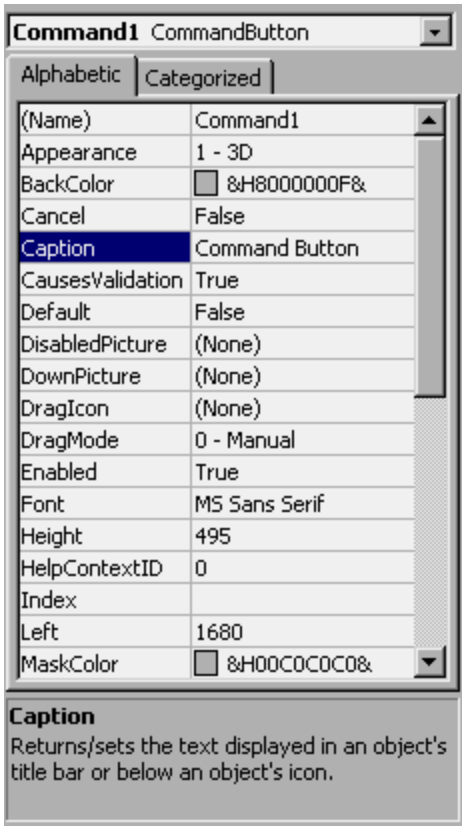
Visual Basic provides us an interface called a **form**. The form is a tool we can use to visually format how the screen will look to our users.



The particular items on the screen, such as drop-down lists, OK and Cancel buttons, Menus, text boxes, and pictures are all referred to as **controls** in Visual Basic. Visual Basic provides nearly 100 controls for you to use to develop your screens, and if that is not enough, there are more than half a million other controls available from other companies that you can use to make your screen.

Example of Controls on a Form: TextBox, CommandButton, and Label





The specific appearance of an individual control is managed through its **properties**. Some properties, such as width and height can be set using the mouse, other properties can be set using the property window, as shown here.

After your screen design is complete, you are ready to run the program to see how it looks.

Demonstration: Modifying screen properties at run time.

Writing Code in Visual Basic

How do we make the program do something?

In all windows programming languages, including Visual Basic, we need to know when the user has done something to the screen so that we can respond to it. In Visual Basic this happens through **events**. An event is something that we can tell our program to respond to. For example, when the user clicks on the OK button, an event is raised and we, as Visual Basic programmers, can tell our program what to do in response to that event. Each control has certain events built into it that it can respond to. Some events include click, mousemove, keypress, datachanged, lostfocus, and gotfocus.

How do we control the flow of code in the program?

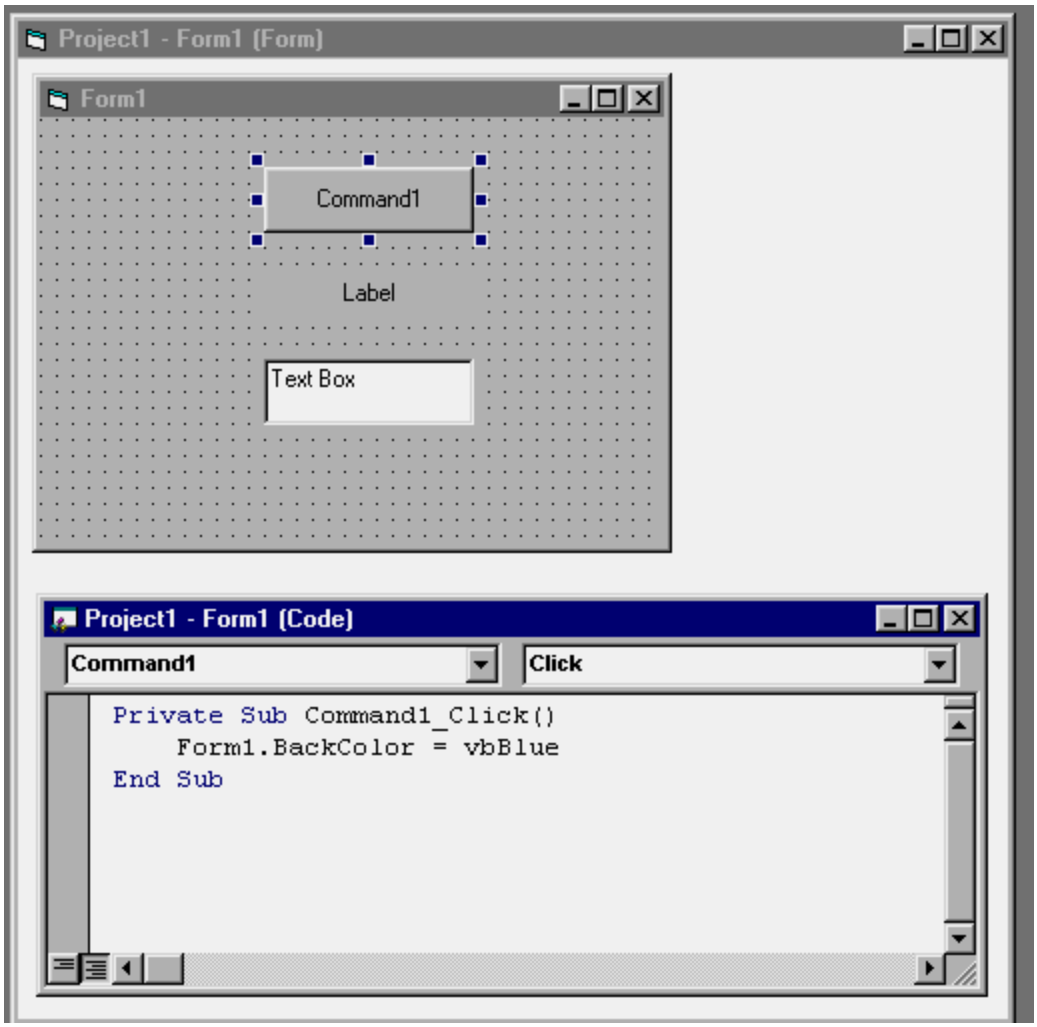
The simple answer is we can't. Visual Basic, and most windows programming languages, are considered to be **Event-Driven Programming Languages**. This means that the user is in control of what will happen next. It is the user who decides whether they will type data into a next box next, or click on a button.

How do we know which events to write code for?

In Visual Basic, you write code only for the events you want to respond to. For example, you might write code that does something when a user types data in a text box (the Change event), but you would not write code for some of the other events available for the text box (such as the click event).

When thinking about writing Visual Basic code, just pretend that you are writing a whole bunch of separate little programs. One program that will run when the user clicks on the OK button, one program that runs when the use selects a menu item, and a third program that runs when the user types data into a text box.

Example: Change the form color when a button is clicked.



Delivering the program to someone else

How do I make it so that other people can use my Visual Basic Program?

At this point, you can give your Visual Basic source code to another person that has installed Visual Basic on their PC, or you can **create an executable** (.exe) and install that on their PC.

Demonstration: Making an executable and running it.

Making things happen on the screen

How do I make the program display data on the screen?

Visual Basic programming is all about modifying properties. Every visual change on the screen is created by changing a property. Just as we can change properties such as background color, font size, text and location at design time, we can modify the same properties at run time to display different data to our users.

Demonstration: Modifying screen properties at run time.

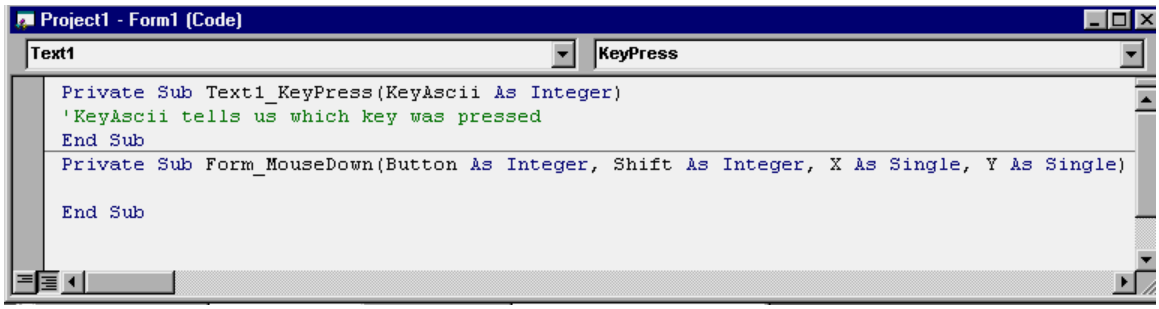
Responding to user input

Tell me more about events!

When thinking about writing a Visual Basic program, you should be aware that hundreds of events may be occurring every second that the program is running. We don't want to respond to all the events, for example, if the user moves the mouse around on the screen we don't want to write code to respond to the MouseMove event (but we could if we wanted to).

Windows tells our Visual Basic program about all the events that are happening, and we write code to respond to only the events we are interested in.

For some events, windows tells us more than just that the event fired. For example, when a key is pressed on the keyboard, windows not only tells us that a key was pressed, it tells us exactly which key. It does this by passing a parameter in the event. When the user clicks a mouse button, windows not only tells us that a mouse button was pressed, it tells us which button and also gives us the x and y coordinates of the mouse on the screen.



MapObjects with Visual Basic

What about GIS?

As mentioned earlier, there are over a half million other controls that you can use to develop your Visual Basic program. Controls for GIS applications are provided by ESRI.
ESRI provides a lot of controls such as MapObjects. Placing a MapObject control on your form makes it easy to present GIS data from databases and shape files.
Of course, it is up to you to obtain the data that you want to have displayed in the controls.
A lot of data is available from ESRI and other sources to help you get started.

Demonstration: MapObjects Example Program - Identify.vbp.

Thinking about Objects

How do I manipulate the MapObjects in my code?

Thus far, we have focused on the graphical elements of Visual Basic, but there is really a lot that goes on behind the scenes. With VB, we read and write data to and from files and databases, and we read and write data to and from controls on the screen.
Becoming a proficient VB programmer requires that we understand the concepts of **Object**, **Properties**, and **Methods**.
Objects in VB are the things we can manipulate, like a **database**, or a **spatial map**.
Properties are the pieces of information stored about objects such as the **size** and **name**.
Methods are the actions we can perform on the object such as **reading data** from a database, or **adding a layer** to a map.

MapObject Examples

How can I get started with MapObjects?

The applications demonstrated here were downloaded from the ESRI web site at http://www.esri.com/software/mapobjects/index.html. They have other software that you may find useful, as do many other sites on the web.

Demonstration: MapObjects Example Program - Viewer.vbp.

Details about shipping a compiled program (.exe) to your users.

Is shipping my VB program really that easy?

NO. For your users to run your program they will need more than just the .exe that you compile. Visual Basic programs require a number of other runtime files to execute successfully. The number of files and the dependencies between them is so complicated that you will probably need to create an Installation program for your users.
Fortunately, VB provides a **Package and Deployment Wizard** that makes it easy for you to distribute your applications.
This simple diagram shows how various files go into creating what is sent to your users.

```
-----------------------------------------------------|
.vbp------>.frm----form                              |
              --------event code                      |-------------------------------|
           >.frm-----form                             |=====> vb.exe                  |
              --------event code                      |                               |
           >.bas-----shared code                      |                               |===========>Send to users
-----------------------------------------------------|                               |
           >.ocx-----molt10.ocx                       |=====> molt10.ocx              |
-------------------------------------------------------------------------------------|
```