

Hello World Program

In this chapter, you will learn how to use the following Visual Basic Application functions for AutoCAD to World Class standards:

- 1. Opening the Visual Basic Editor in AutoCAD**
- 2. Beginning a New Visual Basic Application**
- 3. Laying Out a User Input Form in Visual Basic**
- 4. Insert a Label into a Form**
- 5. Insert a Textbox into a Form**
- 6. Insert a Label into a Form to Post an Output**
- 7. Insert Command Buttons into a Form**
- 8. Adding a Copyright Statement to a Form**
- 9. Adding Comments in Visual Basic to Communicate the Copyright**
- 10. Declaring Variables in a Program with the Dimension Statement**
- 11. Setting Variables in a Program**
- 12. Using a Label to Communicate with Variables**
- 13. Ending the Program**
- 14. Running the Program**

Open the Visual Basic Editor in AutoCAD

To open the Visual Basic Editor in AutoCAD is essential to creating the program to automate the drawing process. In this version of the World Class CAD – Visual Basic Applications for AutoCAD, we are using AutoCAD 2006, but we just finished using all the programs in this text with a group of programmers using AutoCAD 2002. Their drawings were automatically made just as efficiently as if they were using the most recent version of the Autodesk software.

Select Tools on the Menu bar, then Macro and then Visual Basic Editor. Look to the right of the phrase, Visual Basic Editor and the shortcut keys Alt – F11 is noted. For quick launching of the editor, press Alt – F11

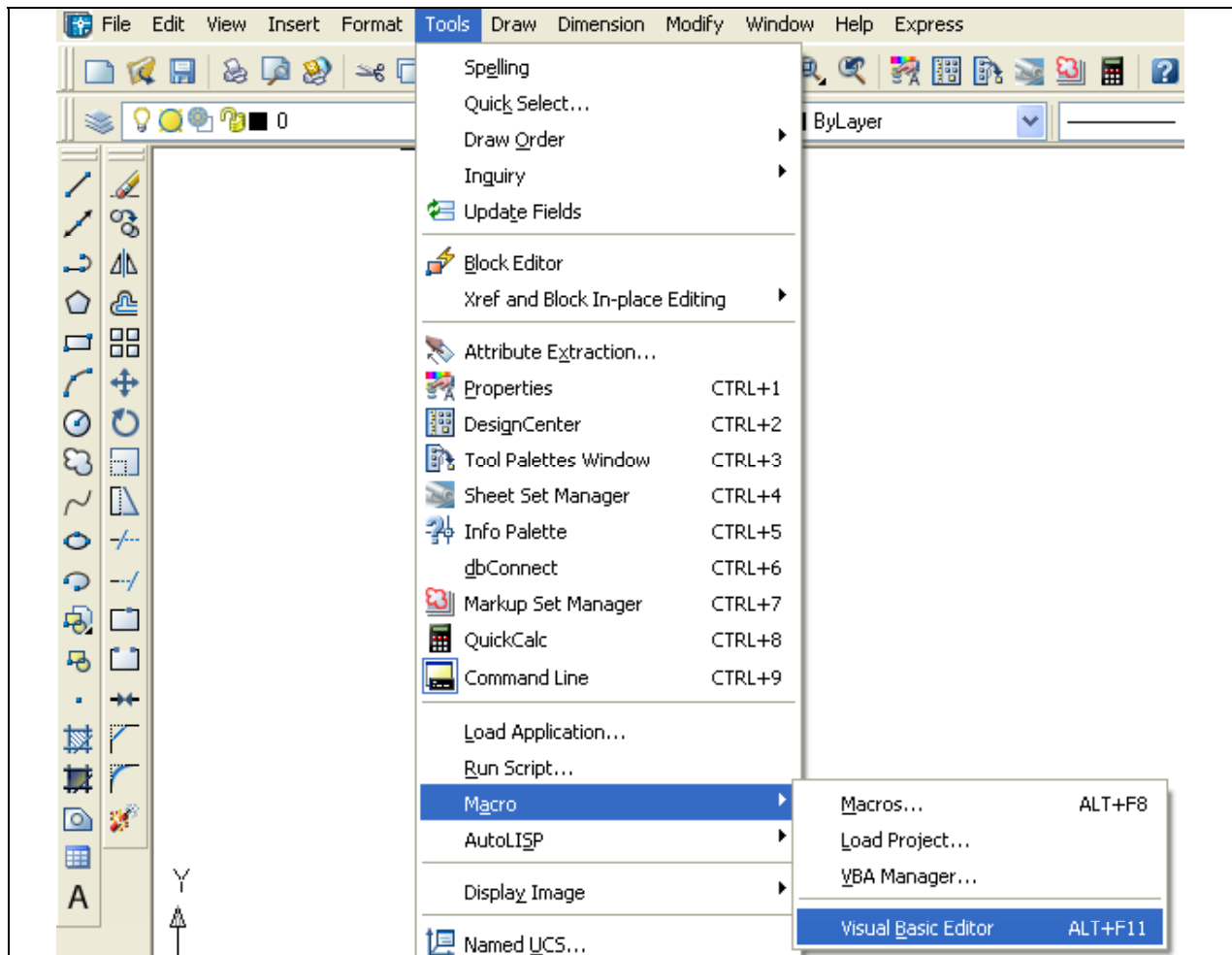


Figure 2.1 – Launching the Visual Basic Editor

The Visual Basic Editor will appear on the computer desktop as a new program application. Looking down on the computer's Task bar, we can see the AutoCAD and Microsoft Visual Basic Editor program tabs. Just single click either program tab to switch between any applications. However, if we close the AutoCAD drawing, unlike a stand alone version of Visual Basic, the Visual Basic Editor will close also.

For those individuals with previous Visual Basic experience, the Visual Basic Editor in AutoCAD has the same layout. The menu bar contains tools for our use as well as the four toolbars shown in Figure 2.3, which are Standard, Debug, Edit and Userform. Presently, only the Standard toolbar is showing. On the left side of the workspace is the Project menu, which shows the files pertaining to this project. Below the Project menu is the Properties pane. If we remember the Properties tool in AutoCAD, using this device will be simple.

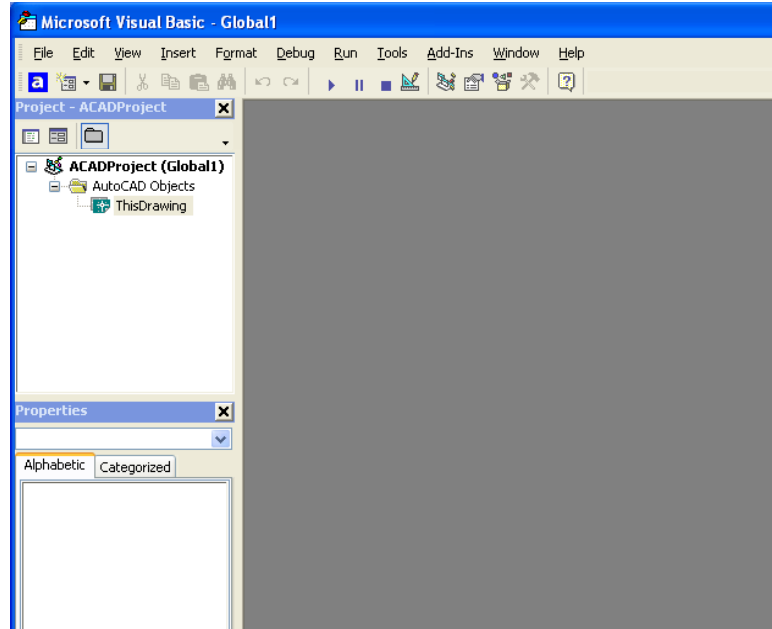


Figure 2.2 – The Visual Basic Editor

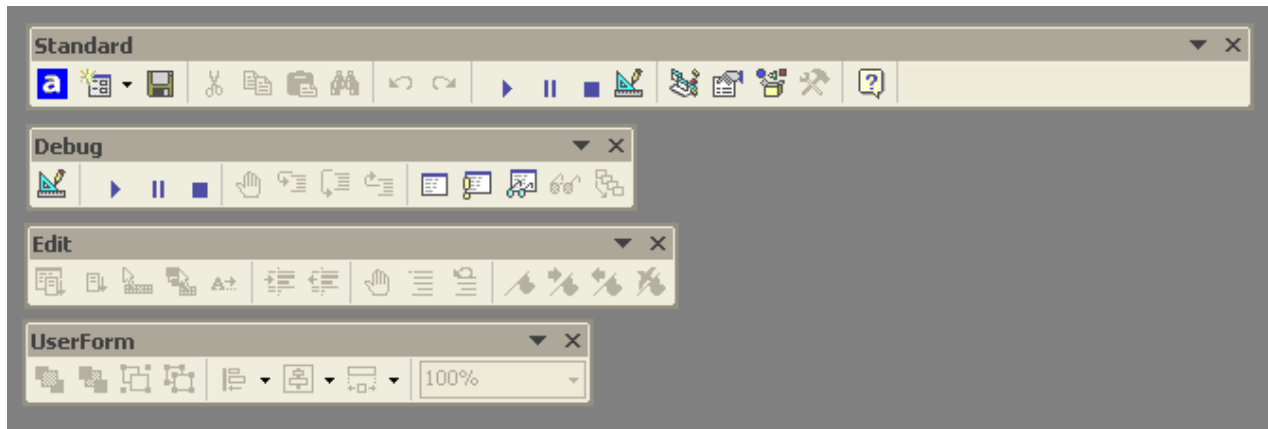


Figure 2.3 – Toolbars in the Visual Basic Editor

With the Visual Basic Editor open, select **File** on the Menu Bar and select **Save Project**. We will create a folder on either the desktop or the My Documents folder called “VBA Programs”. After creating the folder, name the file “Hello World”. The file extension *dvb* as can be easily seen means DCL and Visual Basic programs.

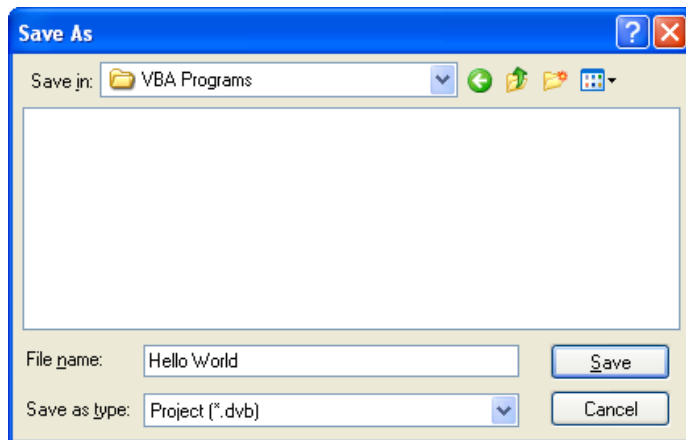


Figure 2.4 – Saving the Hello World Program

Beginning a New Visual Basic Application

Remember, that all programming projects begin with one or more sketches, with one portraying the part, detail, or assembly and the other being the user input form. In this program, Hello World, we will just be running a user input form inside the AutoCAD application, so we only need to sketch the structure of this special dialogue box. In our first project, we will name the input form, **Hello World**. We will place a textbox close to the top of the form to type a name. Over the top of the textbox, we will insert the label, “**Type your name**”. We will have three command buttons, **Hello**, **Reset** and **Exit**. On the bottom of the form, we will write the copyright statement using another label. On this presentation, we can help ourselves by being as accurate as possible, by displaying sizes, fonts, colors and any other specific details which will enable us to quickly create the form. From the beginning of inserting the form into the project, we need to refer to our sketch.

We should train new programmers initially in the art of form building. When using the editor, we insert and size the form, and selecting the Controls Toolbox, we will place all the various input tools and properly label them. Whenever we place an input tool, the properties window will display a list of every attribute associated with the tool, and we will take every effort to arrange the tool by performing such actions as naming, labeling and sizing the visual input device.

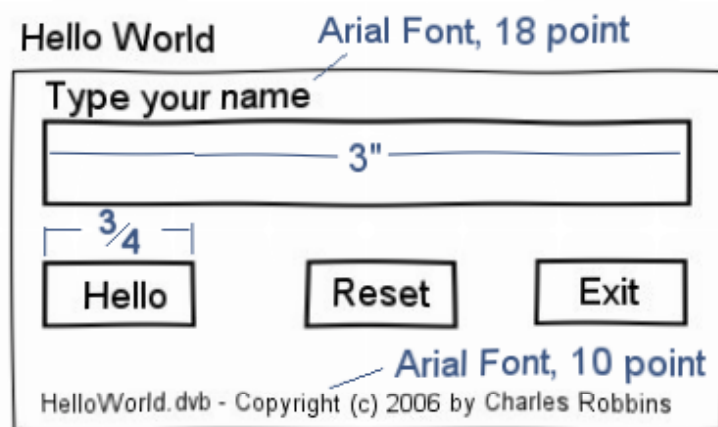


Figure 2.5 – Sketch of the Hello World Form

Now that we have an idea of what the dialogue box in our program will look like, select the **Insert UserForm** button on the Standard toolbar to insert a new form. Instantaneously, the once grey work area is changed to contain our UserForm1. A Form folder with Userform1 is now in the Project menu and the Properties pane contains the attributes associated with UserForm1. (See Figure 2.7)

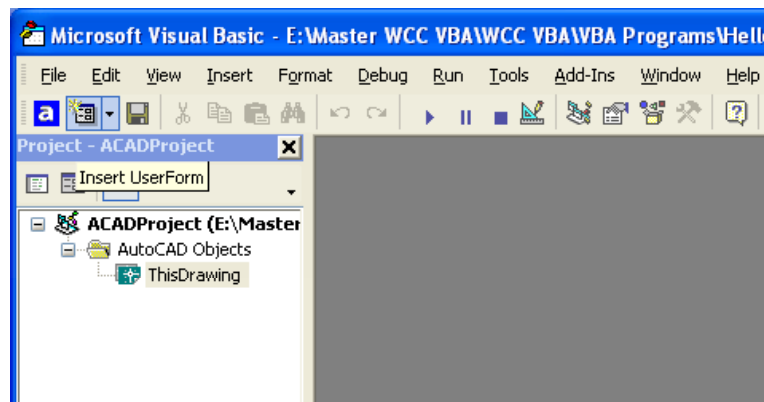


Figure 2.6 – Inserting a User Form

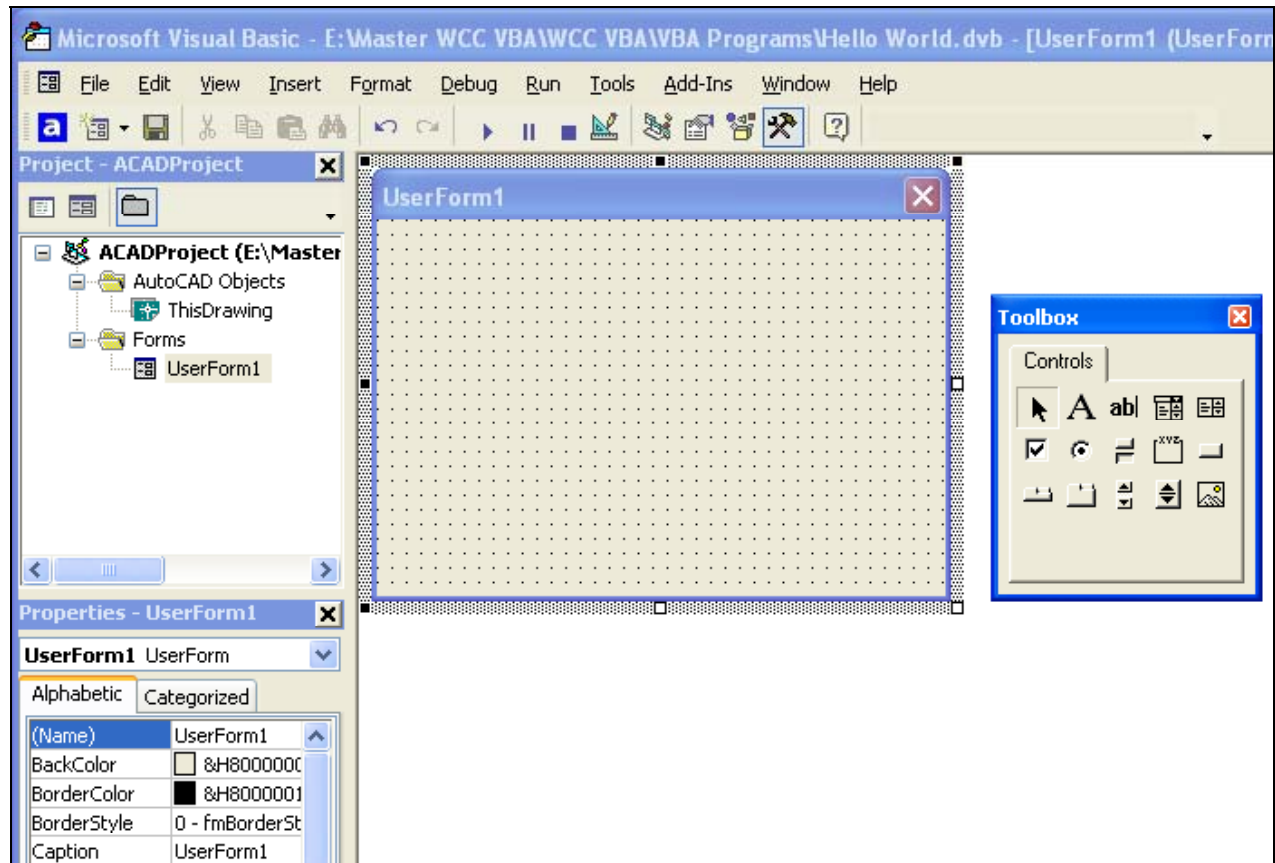


Figure 2.7 – Designing the Hello World Form in Visual Basic

Laying Out a User Input Form in Visual Basic

We will change the **Caption** in the Properties pane to Hello World to agree with the sketch in Figure 2.5. Go ahead and change the form in two other aspects, Height and Width.

Alphabetic	
BackColor	&H80000013&
Height	200
Width	320

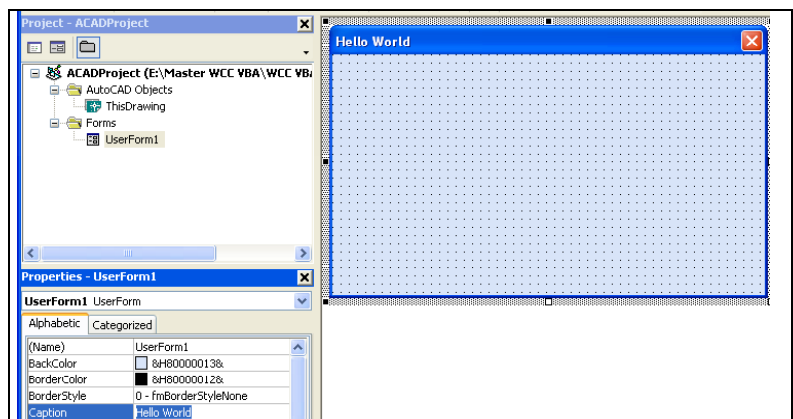


Figure 2.8 – Setting the Caption and other Properties

The form will change in size to the height and width measurement. The background color will change to a light blue. There are many more attributes in the Properties pane that we will use on future projects.

Inserting a Label into a Form

A good form is easy to figure out by the user, so when we are attempting to provide information on the window that will run in AutoCAD; we add labels to textboxes to explain our intent. Press the Label (A) button on the Control Toolbar to add a label. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted label box as shown in Figure 2.9.

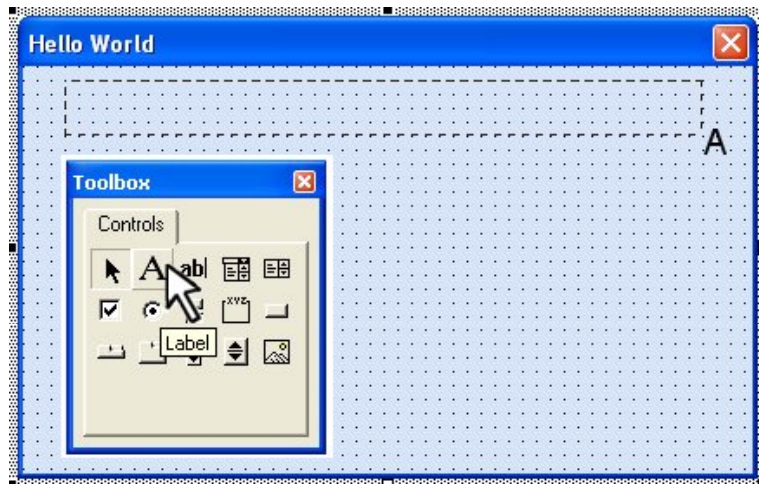


Figure 2.9 – Placing a Label on the Form

We will name the Label using a common Visual Basic naming convention where the programming object is a three letter prefix followed by the name or phrase of the tool. For our first label, the name is **lblName**.

Alphabetic	
(Name)	lblName
BackColor	&H80000013&
Caption	Type your name:
Font	Arial

On the sketch, the label’s caption is “**Type your name:**” The font on the sketch is 18 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three dotted button to open the Visual Basic Font window.

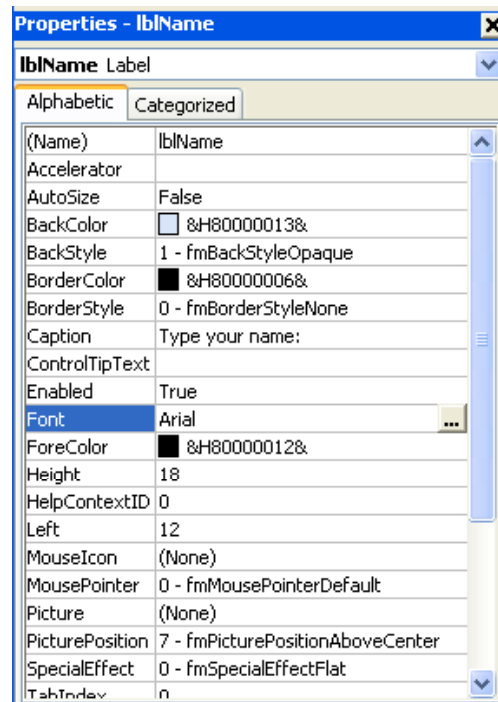


Figure 2.10 – Changing the Font to Arial

We will select the Arial font, Regular font style and 18 size for this project to agree with the initial sketch if the user input form. When we adjust the attributes for the label, these changes do not alter globally for the other objects on the form. If we wish to underline the text or phrase in the label, add a check to the Underline checkbox in the Effects section of the Font window. When we finish making changes to the font property, select the OK command button to return to the work area.

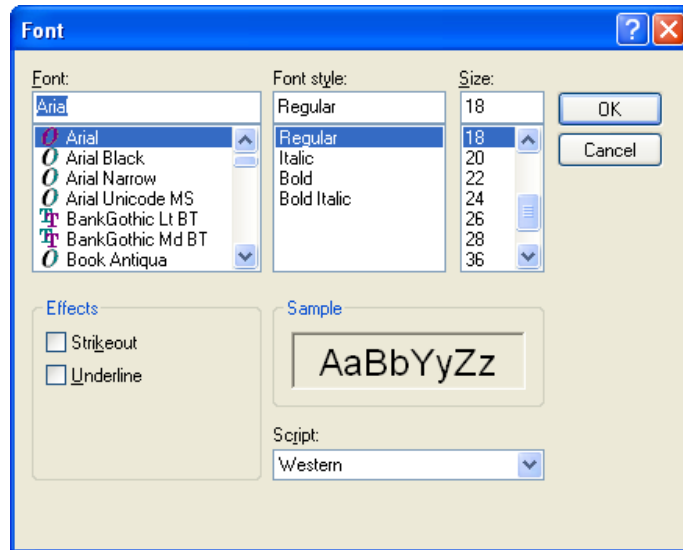


Figure 2.11 – The Font Window in Visual Basic

When the first label is done, the background color of the label matches the background color of the form. In many cases that effect is visually pleasing to the eye, versus introducing another color. Both color and shape will direct the user in completing the form along with the explanation we place on the window to guide the designer in using the automated programs. Use colors and shape strategically to communicate well.

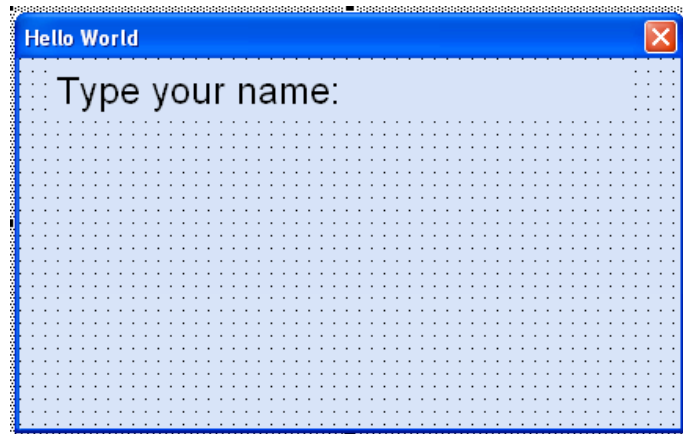


Figure 2.12 – The Finished Label on the Form

Inserting a Textbox into a Form

A textbox is used so that a user of the computer program can input data in the form of words, numbers or a mixture of both. Press the TextBox (ab) button on the Control Toolbar to add a textbox. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted textbox.

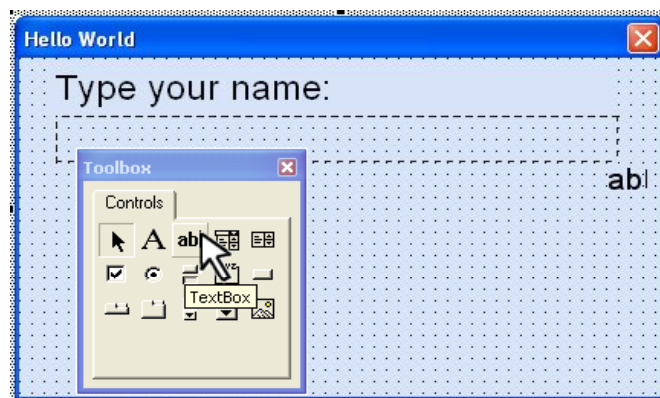


Figure 2.13 – Placing a TextBox on the Form

We will name the TextBox using the three letter prefix followed by the name or phrase of the tool. For our first textbox, the name is **txtName**.

Alphabetic	
(Name)	txtName
Height	24
Width	276

The font on the sketch is 18 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three dotted button to open the Visual Basic Font window. Make the changes like we did on the Label and press OK to save the property.

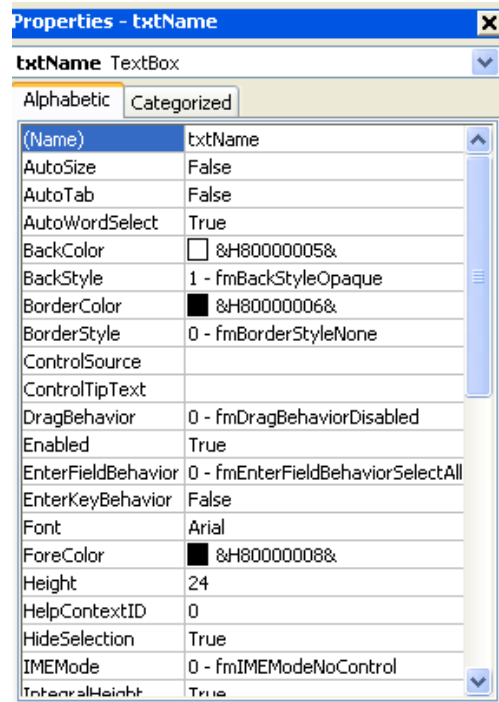


Figure 2.14 – Changing the (Name) to txtName

Inserting a Label into a Form to Post the Output

Some labels on a form are in a position to display an answer after the user inputs data and presses the command button to execute the application. To add this label, press the Label (A) button on the Control Toolbar to add a label. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted label box as shown in Figure 2.15.

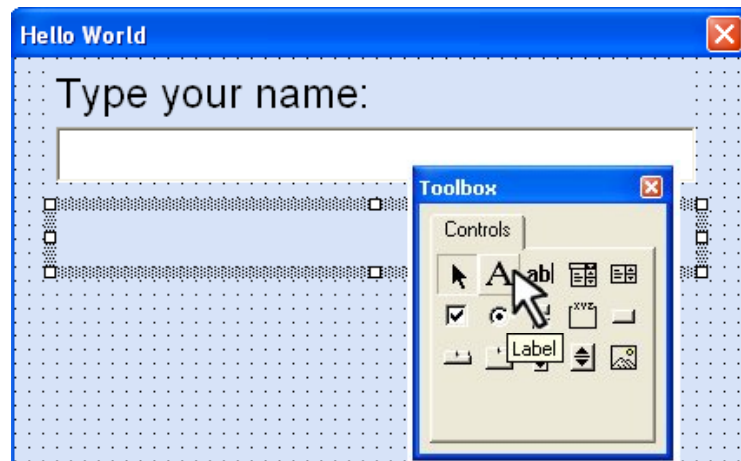


Figure 2.15 – Placing another Label on the Form

We will name the Label using the name is **lblGreeting**.

Alphabetic	
(Name)	lblGreeting
BorderStyle	1 - frmBorderStyleSingle
Caption	
Font	Arial
Height	24
Width	276

The font on the sketch is 18 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three dotted button to open the Visual Basic Font window. Make the changes as we did before and press OK to save the property.

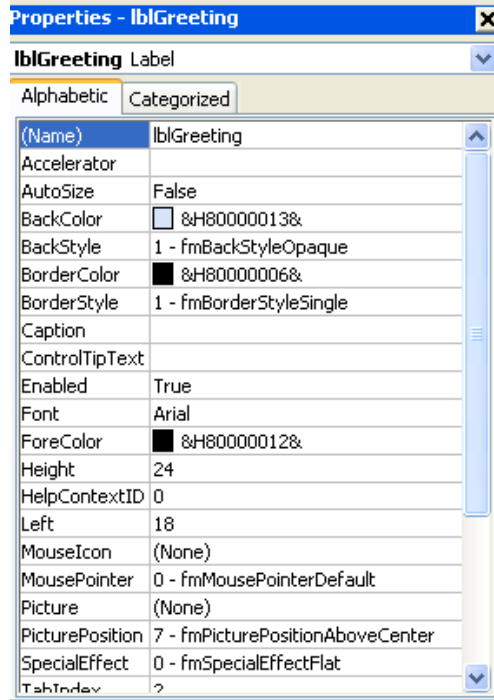


Figure 2.16 – Changing the (Name) to txtName

Inserting a Command Buttons into a Form

A command button is used so that a user will execute the application. Press the Command button on the Control Toolbar to add a command button. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the command button as shown in Figure 2.17.

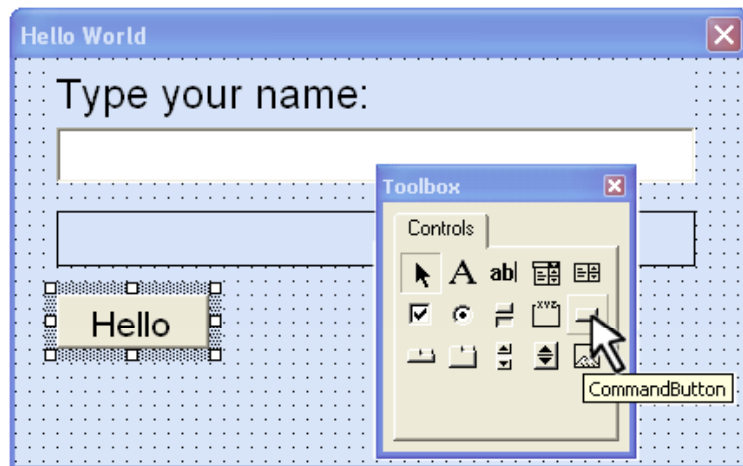


Figure 2.17 – Insert a Command Button onto a Form

We will name the command button using the name is **cmdHello**.

Alphabetic	
(Name)	cmdHello
Caption	Hello
Font	Arial
Height	24
Width	66

The font on the sketch is 18 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three dotted button to open the Visual Basic Font window. Make the changes as we did before and press OK to save the property.

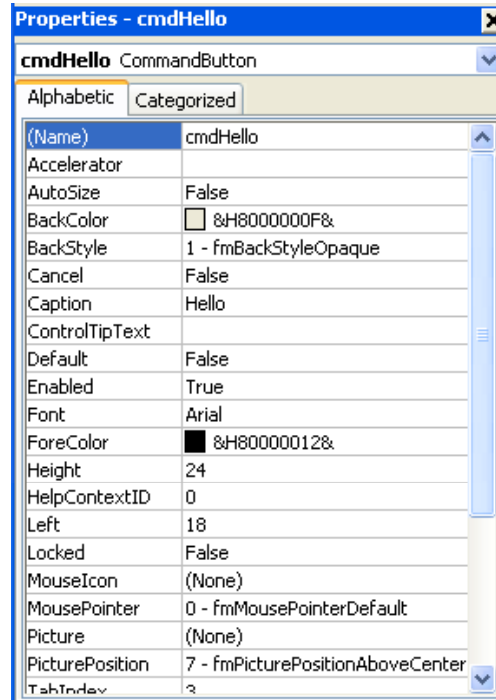


Figure 2.18 – Changing the (Name) to cmdHello

Add a second Command button, named cmdReset is for clearing the txtName and lblGreeting objects. The third command button is to exit the program. When the user presses the Exit command button, the application closes and full control of the manual AutoCAD program returns to the user. Notice the equal spacing between the command buttons gives a visually friendly appearance.

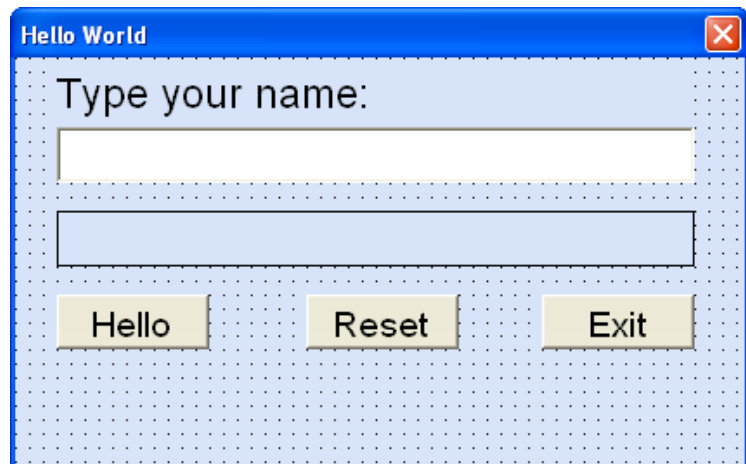


Figure 2.19 – Insert Two More Command Buttons

Adding a Copyright Statement to a Form

At the beginning of a new program, we will expect to see an explanation or any special instructions in the form of comments such as copyright, permissions or other legal notices to inform programmers what are the rules dealing with running the code. Comments at the opening of the code could help an individual determine whether the program is right for their application or is legal to use. The message box is a great tool when properly utilized to inform someone if they are breaking a copyright law when running the code.

Finish the form with the following copyright information.

'hello world.dvb copyright (c) 2006 by charles robbins

If there are special rules or instructions that the user needs to know, place that information on the bottom of the form.

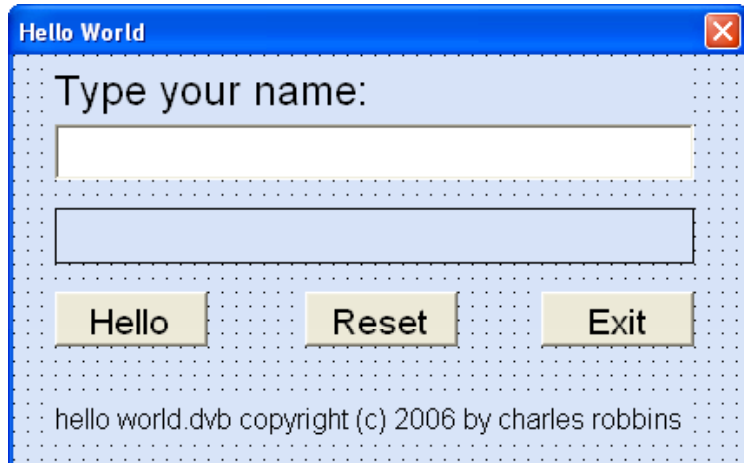


Figure 2.20 – Adding a Copyright Statement

Adding Comments in Visual Basic to Communicate the Copyright

The comments we placed in the first three lines of the program will inform the individual opening and reading the code, but those user that may run the application without checking, the label on the bottom of the form with the copyright information is a great tool to alert the client to the rules of the program and what will the application do.

To begin the actual coding of the program, double click on the Hello command button. At the top of the program and before the line of code with Private Sub cmdHello_Click (), place the following comments with the single quote (') character. Remember, the single quote character (') will precede a comment and when the code is compiled, comments are ignored.

Type the following line of code:

'Hello World.dvb copyright (c) 2006 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user to type their name
'When the user clicks on the Hello button, a greeting, with the date and time is given

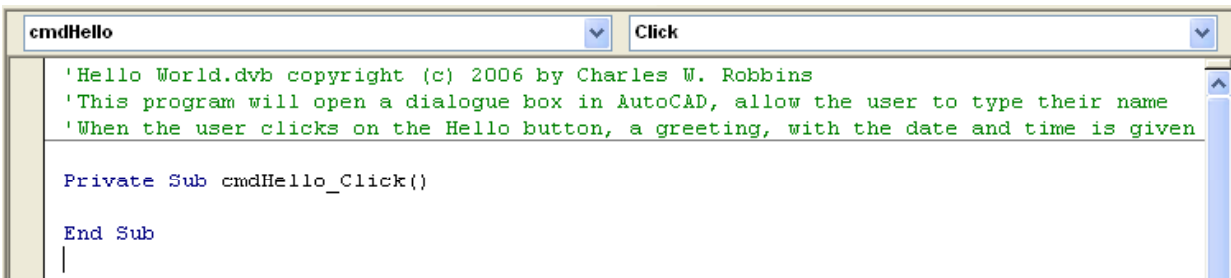


Figure 2.21 – Adding a Copyright Statement

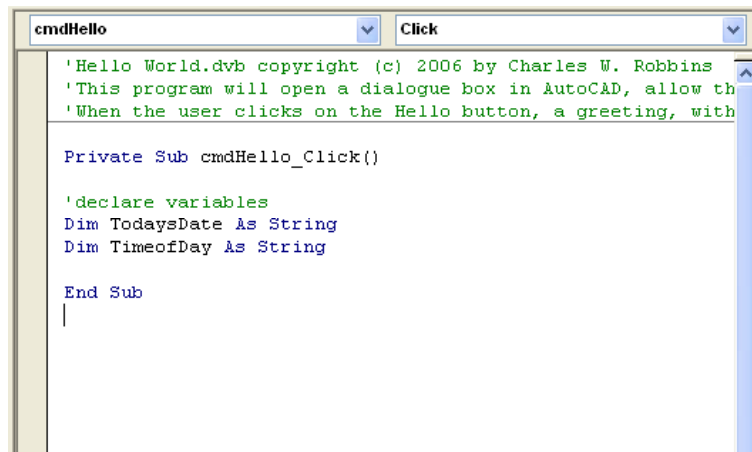
Declaring Variables in a Program with the Dimension Statement

When we are going to use a number, text string or object that may change throughout the life of the code, we create a variable to hold the value of that changing entity. In Visual Basic, the dimension statement is one of the ways to declare a variable at the procedure level. The other two ways are the Private and Public statements, which we will use in later chapters.

In our program, we will retrieve the date and time from the personal computer running the application and place the values in variables called **Today'sDate** and **TimeofDay**.

Type the following code:

```
' declare variables
  Dim Today'sDate as String
  Dim TimeofDay as String
```



```
cmdHello Click
'Hello World.dvb copyright (c) 2006 by Charles W. Robbins
' This program will open a dialogue box in AutoCAD, allow th
'When the user clicks on the Hello button, a greeting, with

Private Sub cmdHello_Click()

  'declare variables
  Dim Today'sDate As String
  Dim TimeofDay As String

End Sub
|
```

Figure 2.22 – Declaring Variables with Dim Statements

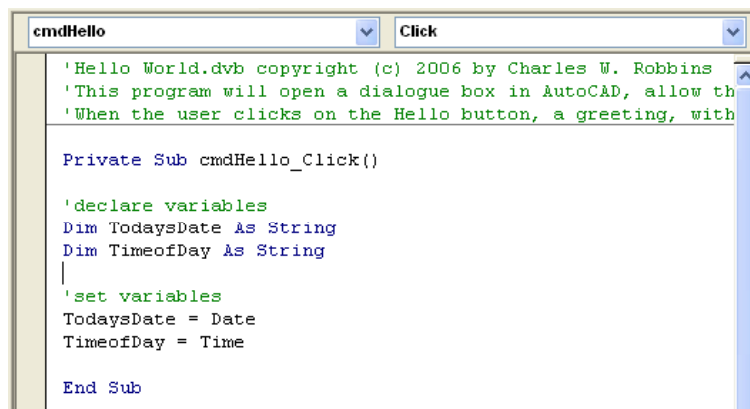
Notice that the variable name should be a word or a phrase without spaces that represents the value that the variable contains. If we want to hold a value of one's date of birth, we can call the variable, DateofBirth. The keywords Date and Birth are in sentence case with the first letter capitalized. There are no spaces in the name. Some programmers use the underscore character (_) to separate words in phrases. This is acceptable, but a double underscore (__) can cause errors if we do not detect the repeated character.

Setting Variables in a Program

Next, we will set the variables using the equal function. First Today'sDate will equal the Date on the calendar in the personal computer. Second, the TimeofDay will equal the Time on the clock on the personal computer.

Type the following code:

```
'Set variables
  Today'sDate = Date
  TimeofDay = Time
```



```
cmdHello Click
'Hello World.dvb copyright (c) 2006 by Charles W. Robbins
' This program will open a dialogue box in AutoCAD, allow th
'When the user clicks on the Hello button, a greeting, with

Private Sub cmdHello_Click()

  'declare variables
  Dim Today'sDate As String
  Dim TimeofDay As String
  |
  'set variables
  Today'sDate = Date
  TimeofDay = Time

End Sub
```

Figure 2.23 – Setting the Variables in the VBA Code

Using a Label to Communicate with Variables

The second message box is more difficult than the first, since we will incorporate the text strings in quotes with the variables. To bring text together or to concatenate the sentence, we will use the & character. When we want a space, we place a space in quotes “ ” or a period “.”

Go ahead and type the following code:

```
'display message
  lblGreeting.Caption = "Hello, " + txtName.Text + ". It is " + TimeofDay + " on " + TodaysDate
```

```
'Hello World.dvb copyright (c) 2006 by Charles W. Robbins
' This program will open a dialogue box in AutoCAD, allow the user to type their name
' When the user clicks on the Hello button, a greeting, with the date and time is given

Private Sub cmdHello_Click()

  'declare variables
  Dim TodaysDate As String
  Dim TimeofDay As String

  'set variables
  TodaysDate = Date
  TimeofDay = Time

  'display message
  lblGreeting.Caption = "Hello, " + txtName.Text + ". It is " + TimeofDay + " on " + TodaysDate
End Sub
```

Figure 2.24 – Computing the Greeting with String Concatenation

The “Hello, ” + txtName.Text is concatenated with the + sign. We can use multiple + signs to add text strings and variables containing text together.

Resetting the Data

To clear the textbox or label containing the greeting, we will set the textbox for Name, txtName.text property to a blank entry by using the equal sign “=” and the null string “”. This makes the property blank. We will set the label for Greeting, lblGreeting.caption property to a blank entry by using the equal sign “=” and the null string “”, and this will make that property blank, also. Notice that after the control object name the dot(.) separates the suffix which is the name of the property for that object.

```
Private Sub cmdReset_Click()
  txtName.Text = ""
  lblGreeting.Caption = ""
End Sub
```

Figure 2.25 – Computing the Reset Button by Clearing a Textbox and Label Caption

Exiting the Program

```
'Hello World.dvb copyright (c) 2006 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user to type their name
'When the user clicks on the Hello button, a greeting, with the date and time is given

Private Sub cmdExit_Click()
'unload and exit the program
Unload Me
End
End Sub

Private Sub cmdHello_Click()

'declare variables
Dim TodaysDate As String
Dim TimeOfDay As String

'set variables
TodaysDate = Date
TimeOfDay = Time

'display message
lblGreeting.Caption = "Hello, " + txtName.Text + ". It is " + TimeOfDay + " on " + TodaysDate
End Sub

Private Sub cmdReset_Click()
txtName.Text = ""
lblGreeting.Caption = ""
End Sub
```

Figure 2.26 – Exiting the Program

To exit this program, we will unload the application and end the program. Type the following code:

```
“Unload and exit the program  
Unload Me  
End
```

Running the Program

After noting that the program is saved, press the F5 to run the Hello World application. The Hello World window will appear on the graphical display in AutoCAD as shown in Figure 2.27. Notice the professional appearance and presentation of information in a clean dialogue box. The color of the background is neither black nor white, which would match the normal graphical display colors used by designers.



Figure 2.27 – Launching the Program

Type your name in the textbox just as we typed the name “Lisa” as shown in Figure 2.28. If we make a mistake, we can type over the text entry or press the Reset command button to clear the textbox. Press the Hello command button and a greeting is displayed like “Hello, Lisa. It is 9:01:33 PM on 12/18/2006” as shown in Figure 2.28. After experimenting with our program, press the Exit command button to exit the application.



Figure 2.28 – Running the Program

If our program does not function correctly, go back to the code and check the syntax against the program shown in Figure 2.26. Repeat any processes to check or Beta test the program. When the program is working perfectly, save and close the project.

There are many variations of this Visual Basic Application we can practice and obtain information from a personal computer. While we are practicing with forms, we can learn how to use variables, strings and comments. These are skills that we want to commit to memory.

*** World Class CAD Challenge 5-1 * - Write a Visual Basic Application that displays a single input form, allow the user to type in their name, and when executed, the program will greet the user with information obtained from the computer.**

Continue this drill four times using some other form designs, each time completing the Visual Basic Application in less than 30 minutes to maintain your World Class ranking.