

VBScript: Hello Program

In this chapter, you will learn how to use the following VBScript functions to World Class standards:

- 1. Download, Install and Open a VBScript Text Editor**
- 2. Beginning a New VBScript Program**
- 3. Adding Copyright Statement to a Program**
- 4. Using a Message Box to Communicate the Copyright and More**
- 5. Declaring Variables in a Program with the Dimension Statement**
- 6. Setting Variables in a Program**
- 7. Using a Message Box to Communicate with Variables**
- 8. Ending the Program**
- 9. Saving the Program**
- 10. Running the Program**

Download, Install and Open a VBScript Text Editor

We will use a script editor from JGSoft, Just Great Software to complete our first program. We will use their EditPad Lite application program to edit our first script.

In our Internet browser software, type <http://www.jgsoft.com/> to view their home page as shown in Figure 2.1. Select the EditPad link to view their editor webpage. Scroll down and select the Free EditPad Lite link. In this webpage, **right** click on the



Figure 2.1 – The JGSoft.com website

Download EditPad Lite, and pick **Save Target As**. Save the program application in the My Documents folder as shown in Figure 2.2.

Close the Just Great Software website and open the My Documents folder. Double click on the SetupEditPadLite file to run the installation application. After installing the program, we will see the script editor appear on the Windows desktop as shown in Figure 2.3.

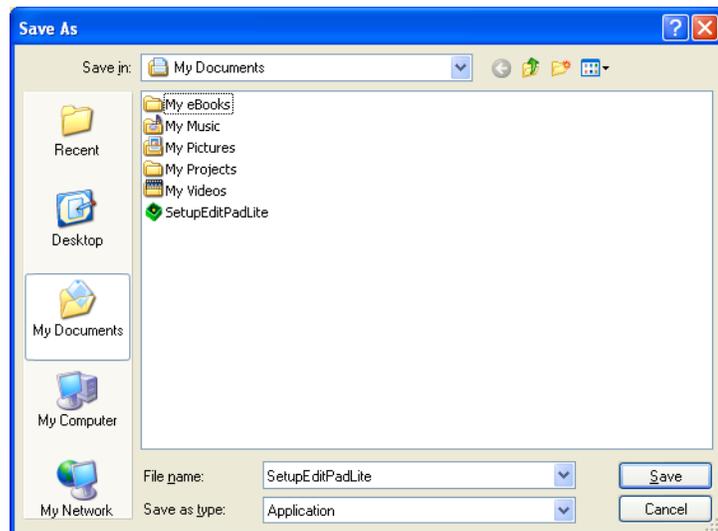


Figure 2.2 – Saving the SetupEditPadLite Application

A script editor will help us to view the program the way we wish and will have additional attributes not found in Microsoft Notepad. The Just Great Software EditPad Lite has some specific features that we list from their EditPad Lite webpage.

- You can open as many files in EditPad at a time as you want.
- You change between the open files by clicking on their tabs.
- EditPad does not impose a limit on the size of files you can open and edit with it. Also, the maximum length of a single line is not limited.
- You will enjoy EditPad's unlimited undo and redo. You can even undo changes after

saving them, as long as you did not close the file.

- EditPad reads and writes UNIX (LF only) and Mac (CR only) text files
- With EditPad you can directly edit Unicode text files and text files encoded in a variety of legacy code pages. EditPad supports all Windows and ISO-8859 code pages, plus a wide selection of DOS, KOI8 and EBCDIC code pages. This means EditPad can edit any text file, whether it was saved on a Windows or Linux computer, an old DOS PC or an IBM mainframe.
- If you run EditPad again when there is already an instance running, the file(s) you wish to edit will be opened by the existing EditPad window. This means there will be at most one EditPad window open, which will save you from a lot of task switching. Block functions: save parts of your text to disk and insert a file in the current text.
- You can specify many print settings: font, margins, headers/footers, etc. and immediately see the effects in the print preview.
- You have the option to keep the EditPad window on top of all other windows.
- When you close an unsaved file, EditPad will either warn you, automatically save the file as you prefer.
- Reopen menu that lists the last 16 files opened.
- EditPad puts an icon in the system tray that remains visible, even if EditPad is closed. This way you have easy and fast access to EditPad, without the need to keep it running all the time. If you do not like this, you can, of course, disable this feature and make EditPad behave like a normal Windows application.
- EditPad fully supports double byte character systems, so you can edit texts written in Far East languages, such as Chinese, Japanese or Korean, as easily as those written in European languages.
- Many settings such as word wrap, line numbering and auto indent can be made for each file type individually. You could activate word wrap but not auto indent when editing a plain text file, and just the other way around when editing source code. You can also define your own file types which will be used to build the filters for open and save dialog boxes.
- Configure EditPad's appearance to your own taste and eyesight. Select the colors, font face and font size that you like. Make text cursor highly visible by selecting its size, color and blinking style. Make the mouse pointer easy to spot by choosing its shape and colors.
- Uppercase, lowercase, invert case and initial caps conversion

There are several more script editors for free or for purchase. We recommend our students to use the free trial download application first, write a handful of scripts and purchase the software program after adequate testing.

Beginning a New VBScript Program

With the EditPad Lite program open, select **File** on the Menu Bar and select **Save**. We will create a folder on either the desktop or the My Documents folder called “VBScripts”. After creating the folder, name the file “Hello.vbs”. The file extension *vbs* as can be easily seen

means Visual Basic Script. We can run a finished VBScript file with the *vbs* extension at the command line by going to the Start button, select Run, and then Browse to locate the script program. When the plain text is compiled at runtime, the program will execute our commands in precise order.

If we want to change the font type or size, select Options on the Menu Bar and pick **Font**. The Font window will appear and we can select the font, font style and size. We will select Arial Narrow, Regular font style and 12 point size for this textbook. We suggest that programmers select text shapes that are easy to read. We are using the narrow style font to eliminate the need for word wrapping.

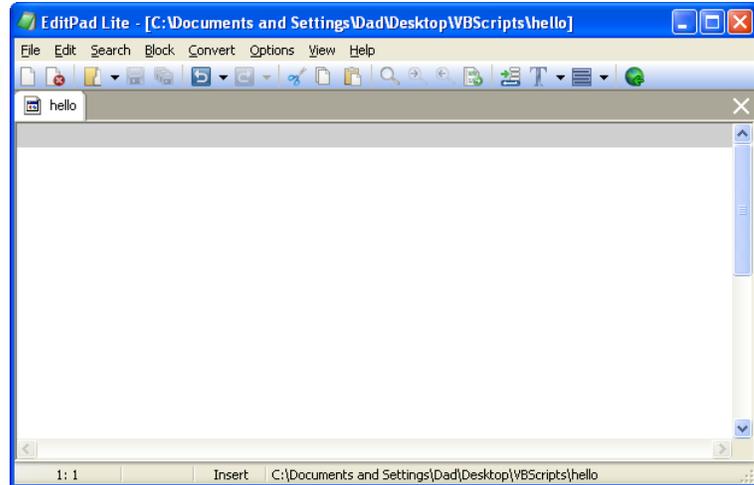


Figure 2.3 – The EditPad Lite Script Editor

Adding a Copyright Statement to a Program

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.

As in Visual Basic, the single quote character (‘) will precede a comment. When the code is compiled, comments are ignored.

Begin the `hello.vbs` script with:

'hello.vbs copyright (c) 2006 by charles w. robbins

Then add a short description for the program:

' this program will open a message box in Windows, extend a greeting, with the date and time

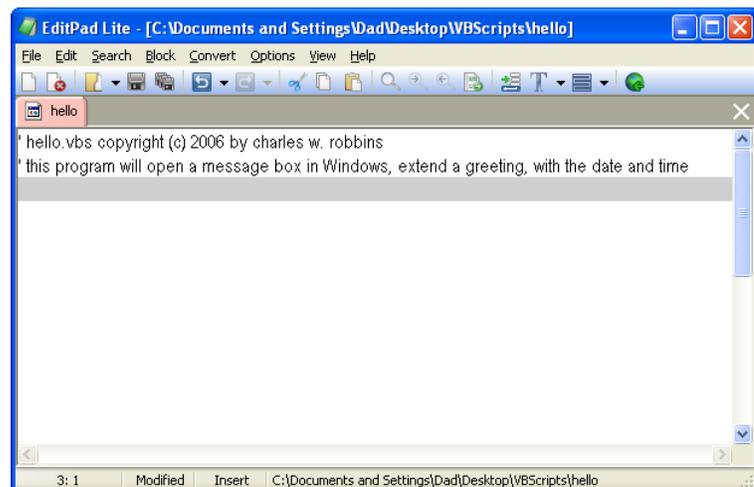


Figure 2.4 – Adding a Copyright Statement

Using a Message Box to Communicate the Copyright and More

The comments we placed in the first two lines of the program will inform the individual opening and reading the code, but those user that may run the application without checking, the message box is a great tool to alert the client to the rules of the program and what will the application do.

In our code add another comment:

' alert the user with a message box

The function MsgBox will launch a message box in Windows. The text or as programmers state "string" will be enclosed in quotes.

Type the following line of code:

```
MsgBox "Hello.vbs copyright  
(c) 2006 by Charles W. Robbins.  
This program will open a message  
box in Windows, extend a greeting,  
with the date and time."
```

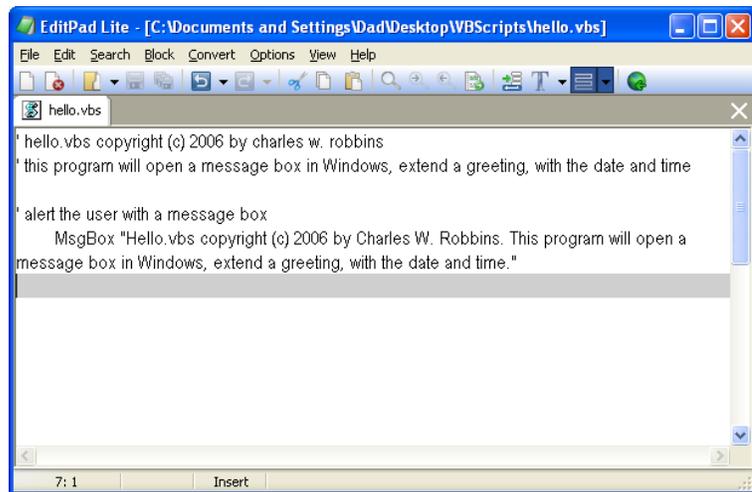


Figure 2.5 – Adding a Message Box

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 script or 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 script and place the values in variables called **TodaysDate** and **TimeofDay**.

Type the following code:

```
' declare variables  
    Dim TodaysDate  
    Dim TimeofDay
```

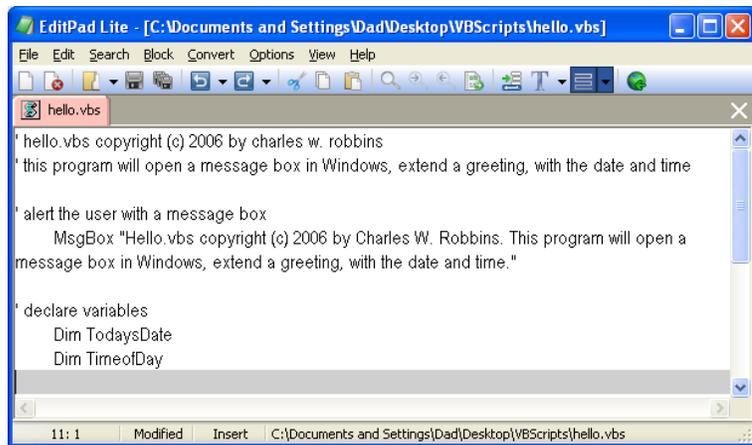


Figure 2.6 – Declaring Variables with Dim Statements

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
```

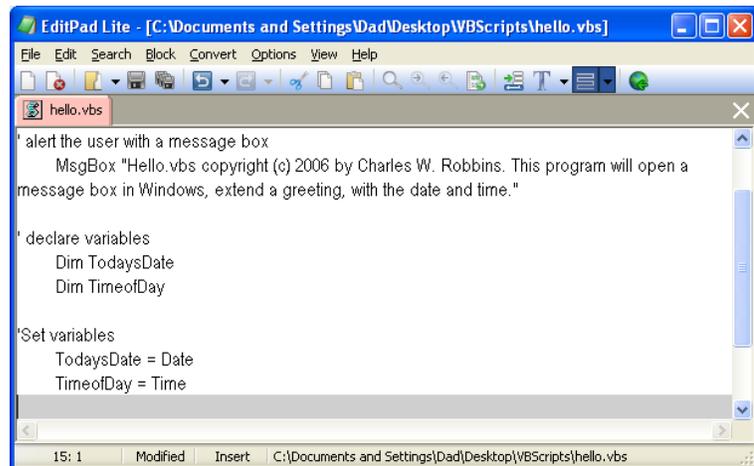


Figure 2.7 – Setting the Variables in the Script

Using a Message Box 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:

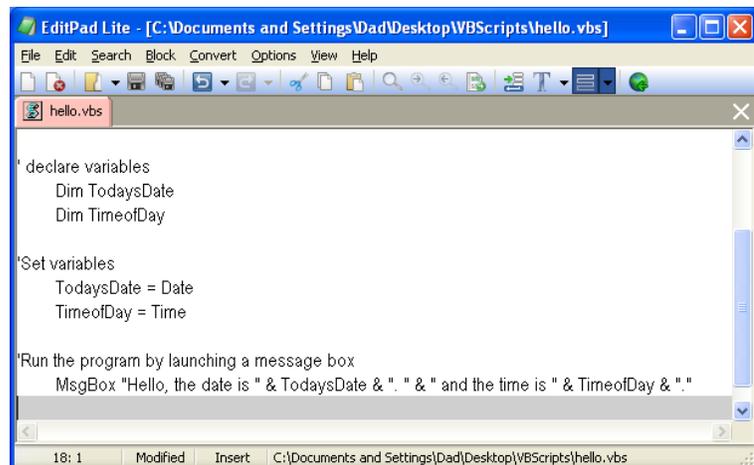


Figure 2.8 – Adding a Message Box with Variables

```
'Run the program by launching a message box
  MsgBox "Hello, the date is " & Today'sDate & ". " & " and the time is " & TimeofDay & "."
```

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

Ending the Program

To end this program, we will type a comment saying so. In the future, this will be more elaborate, but for now we will just get used to announcing the natural divisions of the script.

Type the following code:

'End of program

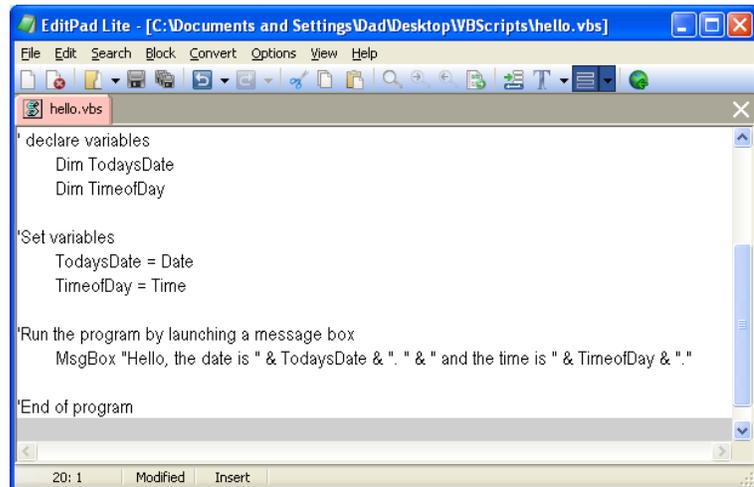


Figure 2.9 – Ending the Program

Programs creating and placing text in a message box are very easy to write once we have achieved writing the first program with these new functions. There are additional exercises for simple routines in the appendixes of this manual. Written below is the entire hello.vbs code for your benefit.

```
' hello.vbs copyright (c) 2006 by charles w. robbins
' this program will open a message box in Windows, extend a greeting, with the date and time

' alert the user with a message box
  MsgBox "Hello.vbs copyright (c) 2006 by Charles W. Robbins. This program will open a
message box in Windows, extend a greeting, with the date and time."

' declare variables
  Dim TodaysDate
  Dim TimeofDay

'Set variables
  TodaysDate = Date
  TimeofDay = Time

'Run the program by launching a message box
  MsgBox "Hello, the date is " & TodaysDate & ". " & " and the time is " & TimeofDay & "."

'End of program
```

Figure 2.10 – The Entire Hello Script

Saving the Program

Now that the program is finished, we need to double check our typing with the text in this manual and then save our program to our folder named “VBScripts”.

Make sure the Save in list box is displaying the VBScripts folder and the File name is “hello.vbs” as shown in Figure 2.11.

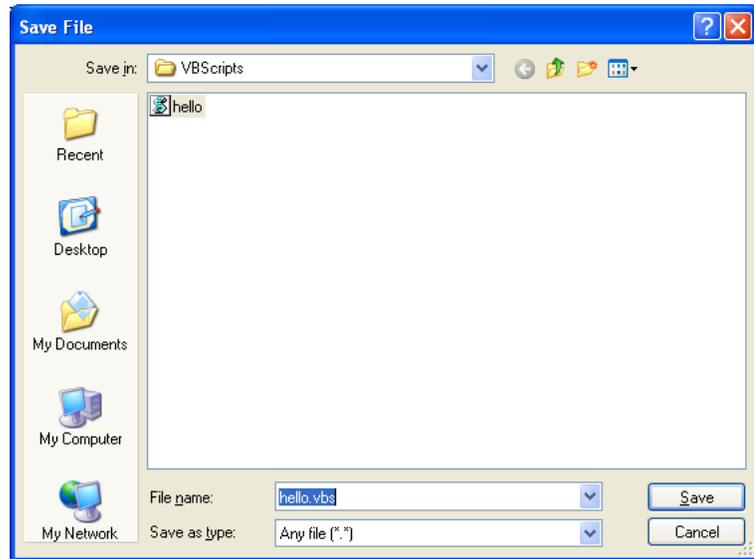


Figure 2.11 – Saving the Program

Running the Program

After noting that the program is saved, press the Start button and pick Run on the Start menu. The Run window will appear on the desktop as shown in Figure 2.12. Select the Browse command button, to open the Browse window.

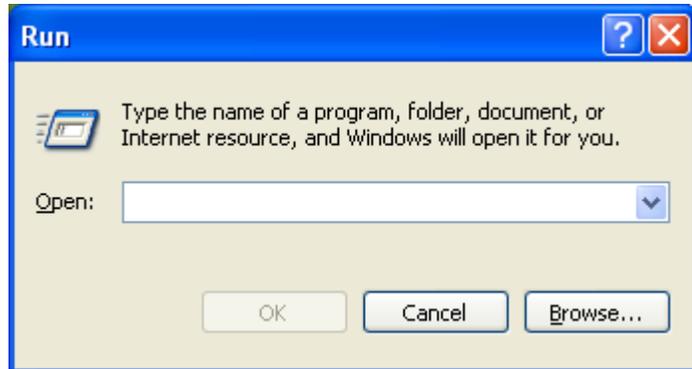


Figure 2.12 – Running the Program

In the Browse window, we need to open the VBScripts folder and select the hello script. If we cannot see the file listed, change the Files of type to “All Files” as shown in Figure 2.13.

Select the Open command button to return to the Run window as shown in Figure 2.14. Press the OK button to execute the script.

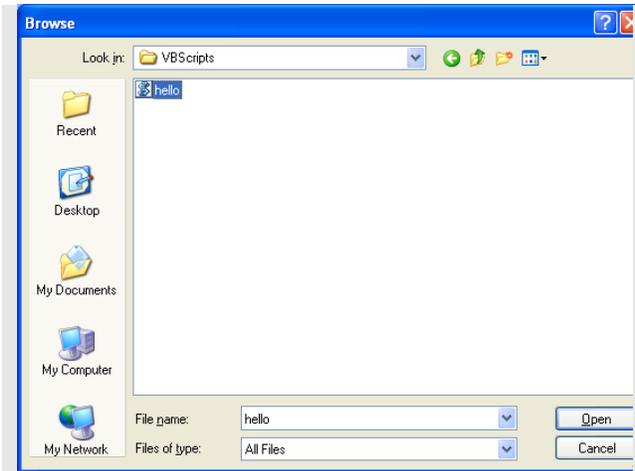


Figure 2.13 – Browse for the Program

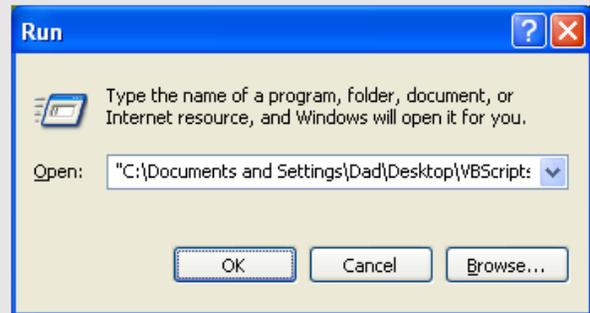


Figure 2.14 – Executing the Program

The first message box to appear will be the regular text message with the program name and the description of the program as shown in Figure 2.15. Press the OK command button to close the message box.



Figure 2.15 – The First Message Box in the Script

The second message will appear with the current date and time read from the personal computer as shown in Figure 2.16. Press the OK command button and the script is done.

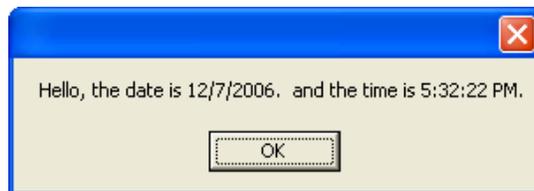


Figure 2.16 – The Second Message in the Script

There are many variations of this script we can practice and obtain information from a personal computer. While we are practicing the message boxes and strings, we learn how to use variables, strings and comments. These are skills that we want to commit to memory.

*** World Class CAD Challenge 9-1 * - Write a Script that displays two message boxes, the first will contain the script name, copyright date and author. The second message will display information from the computer.**

Continue this drill four times using some other messages, each time completing the VBScript in less than 30 minutes to maintain your World Class ranking.