Lesson 9
Creating Spreadsheets and Reports

This lesson will examine how to create spreadsheets and reports. It consists of a pre-lab and four labs.

Lesson 9 Pre-lab Summary

The following are described in Lesson 9 Pre-lab. See Appendix E, page E-59.

- VEE Pro 6: ActiveX References
- Microsoft Excel™ Column and Row Notation
- Microsoft Excel™ Cell Access
- The ActiveX Object Variable
- ActiveX Object Discovery
- Declare Variable

As noted in Lesson 1, Appendix B includes a cross-reference to each of these items and to all objects and subprograms in the labs of this and later lessons.

Overview

Lab 9.1 – Sending VEE Pro Data to an Excel™ Spreadsheet via Globals
This lab will show you how to transfer VEE Pro data to Microsoft Excel™. Declared global variables and function calls are also introduced so you can learn to generate spreadsheets.

Lab 9.2 – Creating a VEE Pro to Excel™ Template
This lab will enable you to create a generic template. It will also allow you to store test data as arrays and to modify the template to fill cells with data in whatever format you may specify.

Lab 9.3 – Using Microsoft Word™ to Prepare VEE Pro Reports
This lab will show you how to send text, a screen-dump of a VEE Pro pop-up panel with an XY display, and a time stamp to a Microsoft Word™ document.

Lab 9.4 – Using VEE Pro to Prepare and Directly Print Reports in Microsoft Word™
This lab will show you how to load data into Microsoft Word™ and automatically print the resulting document.

This lesson will focus upon using ActiveX to access Microsoft Excel™ and Word™ in the preparation of spreadsheets and reports. VEE Pro allows you to document your test data and save it both electronically and as a hard copy. Excel™ and Word™, accessed via ActiveX, are so flexible that you are only limited in its applications to your everyday test documentation by the amount of time that you can devote to developing new and meaningful presentations. It is imperative that you consider the potential reader of your reports and spreadsheets so that you can prepare them in a standard format from your tests.

Lab 9.1 – Sending VEE Pro Data to an Excel™ Spreadsheet via Globals

This lab will show you how to send (transfer) VEE Pro data to Microsoft Excel™. Virtual test data will be generated. Declared global variables and function calls are also introduced so you can learn to generate spreadsheets.

Open your VEE Pro program and

1. Clear your Work Area, deselect the Program Explorer, and maximize Main.

Sending VEE Pro data to an Excel™ spreadsheet via ActiveX

2. Select Menu Bar => Device => ActiveX Automation References…. Figure 9.1 will appear.

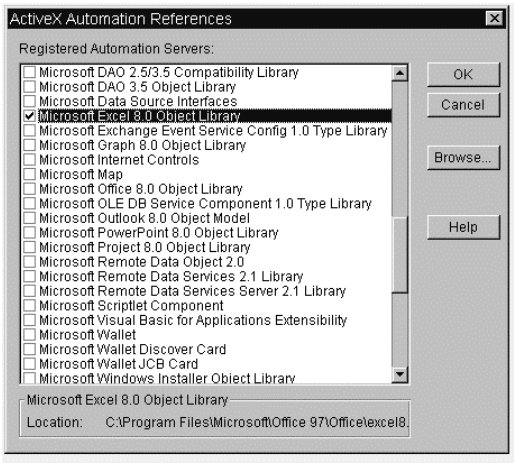

Figure 9.1. The ActiveX Automation References box

3. Select Microsoft Excel™ Object Library; click OK.
   Note 1: This approach may not work with older Windows programs; consult their manual.
   Note 2: This assumes that you have a version of Microsoft Excel™ installed.
4. Select Start on your computer; (lower left-hand corner); run Microsoft Excel™.
   Note 1: At the bottom of your screen, in the Windows Tool Bar, you should now have a long, rectangular icon labeled Microsoft Excel™; return to VEE Pro by selecting the icon: VEE Pro.

Note 2: Microsoft Excel™ will automatically start via ActiveX Automation when the Vehicle Radiator program is run.

5. Select Menu Bar => Device => UserFunction; rename it globals; click OK.
6. Select Menu Bar => Data => Variable => Declare Variable; place it in the top-left corner inside the globals UserFunction.
7. Change the white-space Name to sheet; change Type to Object.
   Note 1: When you click outside this, or any other Declare Variable object, its title-bar name is automatically converted; the word "Variable" becomes the name that you typed into the Name window.
   Note 2: The "Specify Object Type" area can be used select the exact type of Object (such as Excel sheet versus an Excel cell) for error checking. It can also be used to create event callback function to "catch" an event from Excel.
8. Clone this object three times; change the variable names to: app, range, and window.
   Note: Declare Variable allows you to specify the Object Type and Class. This action allows for more specific type checking.
9. Convert all four of them to icons; place them as shown in Figure 9.2; size the globals UserFunction.

Figure 9.2. The globals UserFunction

10. Size the Main window so you can place the globals UserFunction below the Main window; you can now see both on the screen.
11. Save As… this partial program as LAB9-1.
    Note: At the bottom of your screen you should now have two long, rectangular icons labeled VEE Pro – LAB9-1; the other labeled Microsoft Excel™.
12. Select Menu Bar => Device => Call; an object entitled "Call Function" will appear in the Main window; place it in the top-left corner of Main.
    Note: Call globals is technically not required; the Declare Variable objects will activate globals at pre-run.
13. Change its "myFunction" name (in the white space) to globals; click outside the object; its Title Bar will automatically change to Call globals.
    Note: Local User Functions will appear in the Program Explorer Work Area.
14. Convert Call globals to an icon.
    Note: Events allow you to catch, via your VEE Pro UserFunction, events that could occur in an application, such as "right-button-down" in an Excel™ worksheet.
15. Click Device => Formula; place it in the upper center of the Main window.
16. Rename its icon: Set up Excel Worksheet.
17. Connect the Call globals sequence-out (bottom) pin to the "Set up Excel Worksheet" sequence-in (top) pin.
18. Enter, inside "Set up Excel Worksheet", the formula lines as follows; see Figure 9.3.

```
set sheet =CreateObject("Excel.Sheet").worksheets(1);
```

```
set app =sheet.application;
app.visible =1;
set window =app.windows(1);
window.caption =“Test System Results”
```

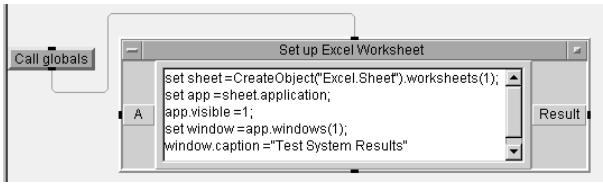

Figure 9.3. Setting up the Microsoft Excel™ worksheet

Note 1: Semicolons are used for expression separators. One expression per line makes the program easier to understand.

Note 2: Properties usually refer to some type of data you either set or get, such as

- get properties via object.property
- set properties via object.property=MaxSize
- call a method via Object.method(parameters)
- get the application property of the sheet object via sheet.application

A method is just a property with parameters. In many cases, a property refers to the attribute of the object (a noun); a method performs an action an the object (a verb). Another difference is: a property (get or set) is very fast to execute; a method requires some time to execute. See Appendix E, page E-xx. Choosing descriptive names for your variables is important.

Note 3: Set is Microsoft syntax used to assign (or set) whatever is on the right-hand side of the assignment operator = to the variable on the left-hand side of the expression.

Example: You have previously declared the variable app as an Object type. Thus you can “set” app to the value on the right-hand side of the expression.

Note 4: CreateObject(“Excel.Sheet”) is located in the Function & Object Browser as a Member of the Built-in Functions Type and the ActiveX Automation Category.

Note 5: GetObject( ) is used to get some data that already exists in a running Excel™ or to load a file into a running Excel™. (A useful reference is the Microsoft™ Office Visual Basic Programmer’s Guide.)

19. Delete the Set up Excel™ Worksheet input pin and output pin.
20. Convert Set up Excel™ Worksheet to an icon.
21. Select Menu Bar => Device => Formula twice.
22. Select Menu Bar => Flow => Repeat => For Range once. Change the For Range object white space as follows:

    From: 1 Thru: 20 Step: 1
23. Rename the Formula objects, connect them, and configure them as shown in Figure 9.4; be certain to delete the “Fill in Title” input and output pins.

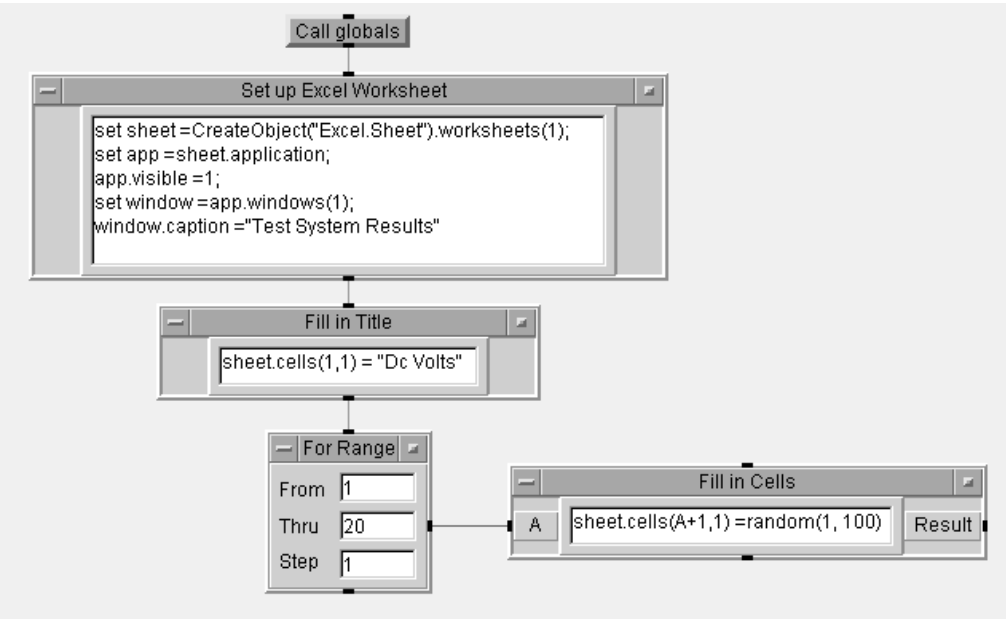

Figure 9.4. Worksheet title and data entries

Note 1: Sheet.cells(1,1) refers to the first row and first column in the Excel™ worksheet. The text "Dc Volts" will be placed there.

Note 2: For Fill in Cells, the variable sheet.cells(A+1,1) gets the row number by adding 1 to the input pin "A" value; it stays in column 1.

Note 3: The value between 1 and 100 returned by "random" is assigned to the specified cell in the worksheet.

Note 4: The For Range object generates the integers from 1 through 20; the "Fill in Cells" places the random number in the specified cell.

24. Get a Formula Object; delete its input terminal; change its formula entries (3) as indicated in Figure 9.5.
25. Get an AlphaNumeric object from the Menu Bar: Display menu; rename it Results Average.
26. Configure and connect all objects exactly as shown in Figure 9.5.

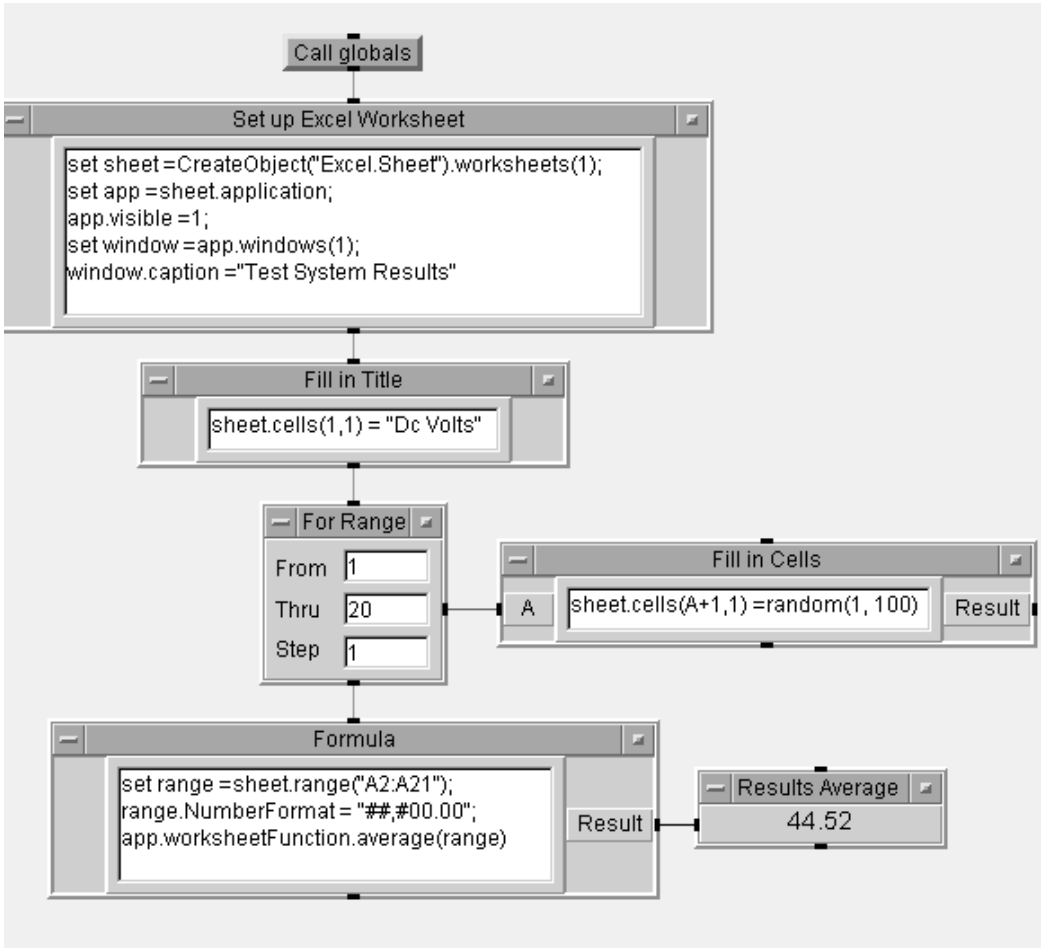

Figure 9.5. The complete Lab 9.1 program

Note 1: The statement set range =sheet.range("A2:A21") sets the VEE Pro variable range to reference the range A2 to A21 on the Excel™ worksheet; "A" refers to the first column in a worksheet.

Note 2: range.NumberFormat = "##,#00.00" assigns the format to each of those cells with the # signs allowing for larger numbers.

Note 3: app.worksheetFunction.average(range) calls an Excel™ function average( ) that returns the average value of the designated range of values; it is displayed in Results Average.

27. Save your program again as LAB9-1.
28. Open the Excel™ program.

    or

    Open the Excel™ program, if necessary, within Microsoft Windows.

Note: Excel™ may open automatically; run this program to determine if it will open automatically before going to the "Start" screen and finding Excel™.

29. Run your program; the Excel™ worksheet appear as noted in Figure 9.6.
    Note: Sheet1 will appear at the bottom of your new worksheet.

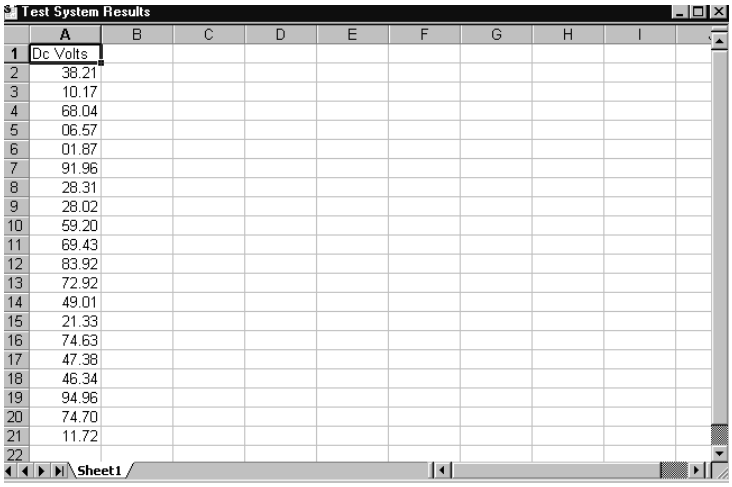

Figure 9.6. The generated Microsoft Excel™ worksheet

◊ 30. Re-run this program several times; switch between VEE Pro and Excel™; the Results Average displayed value will also change.

◊ 31. Change "For Range" to values less than 20; run this program change several times; observe the effect upon the amount of data taken and printed on the spreadsheet.

◊ 32. Change the Formula Object "range.NumberFormat" by increasing the number of zeros to the right of the decimal marker; run this program.

◊ 33 Change the Formula Object "range.NumberFormat" by increasing the number of zeros to at least eight; run this program; note the ability of the spreadsheet column to adapt to these additional decimal values.

34. Close your program without saving it.

Lab 9.2 – Creating a VEE Pro to Excel™ Template

This lab will enable you to create a more generic template keyed to your own work. It will also allow you to store your test data as arrays and to easily modify the template to fill cells with data in whatever format you may specify.

Open your VEE Pro program and

1. Clear your Work Area, deselect the Program Explorer, and maximize Main.

Creating an VEE Pro to Excel™ template

2. Open LAB9-1; change the For Range object to loop ten times; Save As… LAB9-2 immediately.
3. Add an input B to Fill in Cells; alter the statement inside to read
   sheet.cells(A+1,1) = B[A-1]
4. Select => Menu Bar => Click Device => Formula; rename it Array of Test Data.
5. Enter the expression: randomize(ramp(10),4.5,5.5) in Array of Test Data.
   Note: This will create a random array of ten elements with values from 4.5 to 5.5.
6. Delete the input pin of Array of Test Data; change For Range to an icon.
7. Connect the data-output pin of Array of Test Data to the B input of Fill in Cells.
8. Change the range in the Formula box on the bottom of your screen from A21 to A11 which should result in the modified statement reading
   set range=sheet.range("A2:A11");
9. Save your program again as LAB9-2. Your program should look like Figure 9.7.

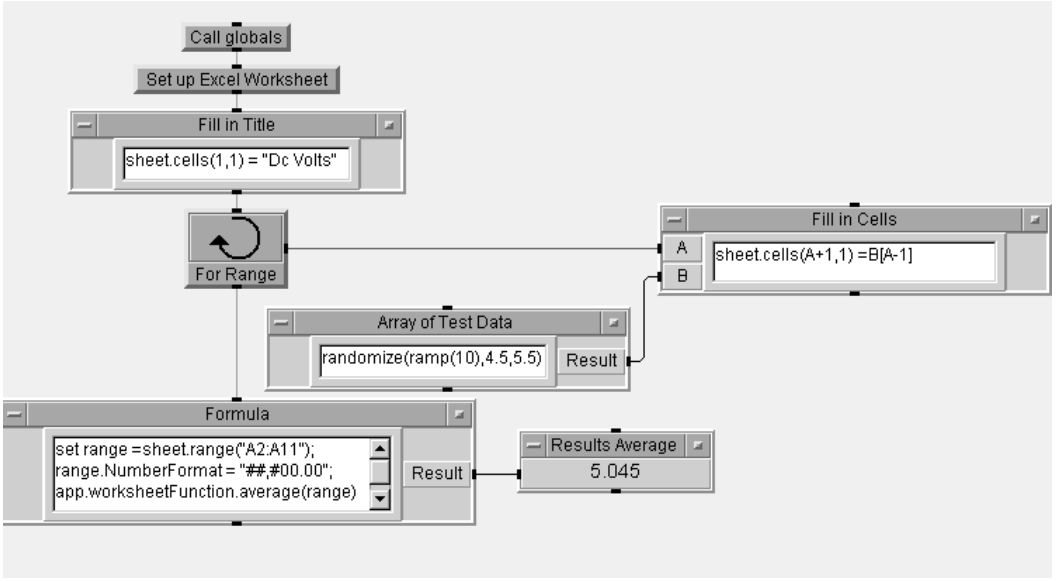

Figure 9.7. The complete Lab 9.2 program

10. Run this program.
    Note 1: Your Excel™ worksheet should look like Figure 9.8 but will display different values.

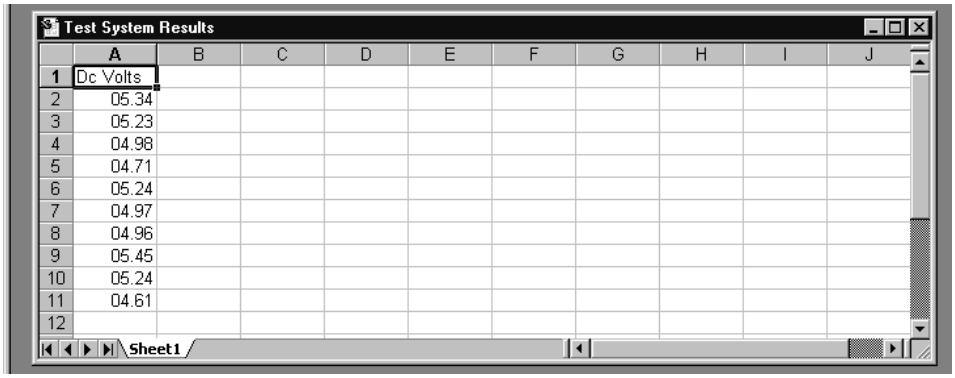

Figure 9.8. Microsoft Excel™ worksheet generated from Lab 9.2

Note 2: For additional methods and properties available in the Excel™ library, go to Menu Bar => Device => Function & Object Browser; Type: ActiveX Objects; Library: Excel. Choose the "Class" you desire to activate. Consult Microsoft documentation for more complete information on these libraries – their classes and their members.

11. Close this program without saving it again.

Lab 9.3 – Using Microsoft Word™ to Prepare VEE Pro Reports

This lab will show you how to send text, a screen-dump of a VEE Pro pop-up panel with an XY display, and a time stamp to a Microsoft Word™ document.

Open your VEE Pro program and

1. Clear your Work Area, deselect the Program Explorer, and maximize Main; Save As… LAB9-3 immediately.

Transferring VEE Pro data into a Microsoft Word™ document

2. Select Menu Bar => Click Device => ActiveX Automation References…; select Microsoft Word™ Object Library, Microsoft Office Object Library, and Microsoft Internet Controls; click OK. (The revision numbers will depend upon the version of Microsoft Office™ you have installed.) See Figure 9.9.

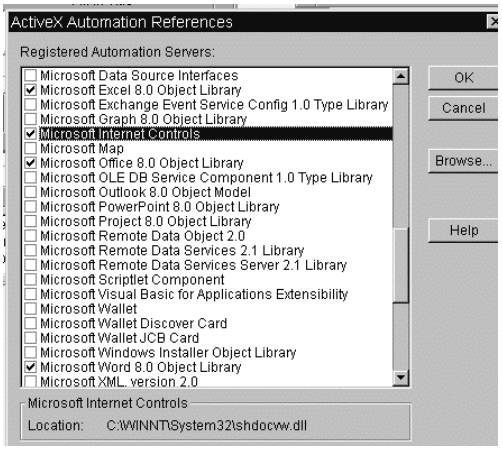

Figure 9.9. The ActiveX Automation References box

3. Select Menu Bar => Click Data => Variable => Declare Variable; change the Type field to Object.
4. Clone Declare Variable four times; name the five Declare Object variables in their "Name" edit field:
   App
   Doc
   Wnd
   Sel
   Bmp

   Note: The Title Bar label will change automatically.
5. Click each Specify Object Type, then click its Edit… button; select the appropriate library and class as follows:
   for App – Library: Word; Class: Application; check the "Enable Events" box
   for Doc – Library: Word; Class: Document; check the "Enable Events" box
   for Wnd – Library: Word; Class: Window
   for Sel – Library: Word; Class: Selection
   for Bmp – Library: Word; Class: Shape

   Note: You cannot access the Enable Events box for the last three objects.
6. See Figure 9.10 to verify your selections.

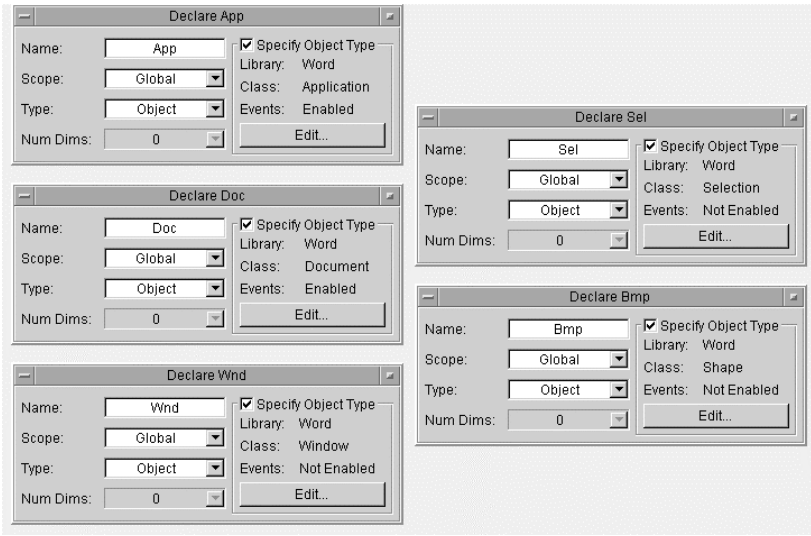

Figure 9.10. The five Declare Variable objects

7. Convert these five Declare Variable objects to icons; place them in a column in the lower-right corner of Main.
8. Select Menu Bar => Device => UserFunction; name it Graph.
9. Select Menu Bar => Device => Virtual Source => Function Generator; place it on the left-center side of Graph.
10. Select Menu Bar => Device => Waveform (Time); place it to the right of Function Generator.
11. Connect the Function Generator to Waveform (Time) Trace1.
12. Highlight Waveform (Time); place the mouse pointer over the window background of Graph; click on the mouse right-hand button.
13. Click Add to Panel; Waveform (Time) will appear in a Panel View.
14. Select Menu Bar => View => Program Explorer; double-click on Main; the Main window will appear.
    Note: An alternative would be to click on the upper-right "_" symbol of Graph so it will appear in the lower-left corner of the screen under Main as an icon.
15. Select Menu Bar => Device => Call; place Call Function in the upper-right corner of Main, click OK.
16. In the Call Graph Function "Name" field, change myFunction to Graph; convert it to an icon. (Its Title Bar will automatically change to Graph.)
17. Select Menu Bar => Device => Formula; place it in the upper-left corner of Main; change its name to **ImageFileName**; delete its input terminal.
18. Change its Formula expression field to: installDir()+"panel.bmp" with no spaces within the quotes.
19. Select Menu Bar => Device => Formula; place it to the right of ImageFileName and under Call Graph.
20. Change the name of Formula to **savePanelImage**; change its formula to SavePanelImage("Graph",fileName,256)

and change its input A pin to read **fileName** with no spaces within the quotes to match the variable name within the expression.

Note 1: Your saved picture will have a depth of 256 colors per pixel.

Note 2: The savePanelImage() function saves the specified UserFunction panel image in either a Windows Bitmap or JPEG format.

21. Select Menu Bar => Device =>Formula; place it below **savePanelImage**; change its formula to:
    Set App = CreateObject("Word.Application")
    and delete its input terminal.
22. Connect these four objects as shown in Figure 9.11; save this program as LAB9-3.

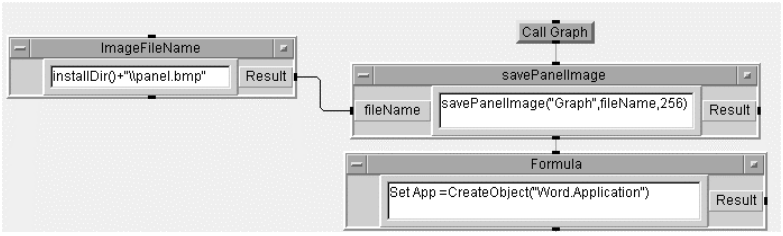

Figure 9.11. Connections for first portion of Lab 9.3

23. Select Menu Bar => Display => Note Pad; enter the following description
    This program is designed to
    stop when Word with the Graph
    appears so you can add your
    text to the Word memo.
    Size the Note Pad around the description.
24. Select Menu Bar => Device =>Formula; change its input-pin name to fileName; delete its output terminal.
25. Enter the formula statements shown in the box of Figure 9.12 below; connect the output and sequence pins as shown.
26. Save your program again

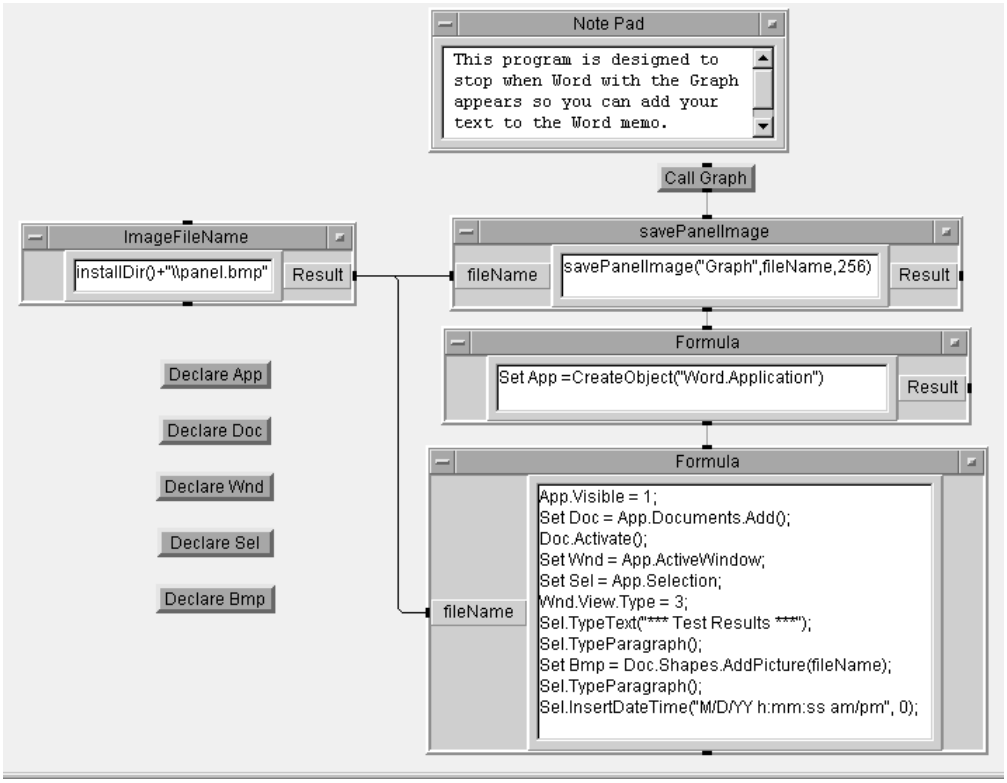

Figure 9.12. The complete Lab 9.3 program

27. Run this program; move the graph downward with the "down-arrow" key so the "Test Results" and date appear above the graph; add whatever statements you prefer. An example is shown in Figure 9.13.

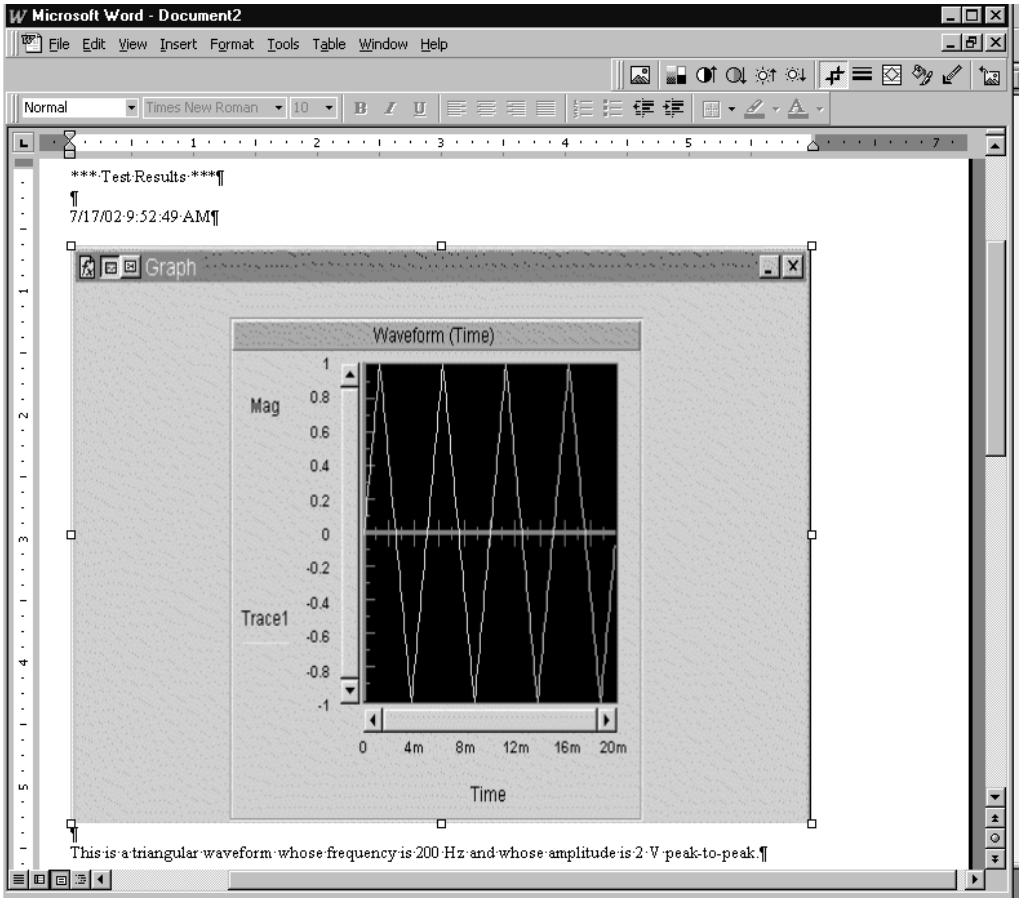

Figure 9.13. A Microsoft Word™ document containing graph and an explanatory statement

◊ 28. Crop your picture so only the Waveform (Time) oscilloscope is displayed by performing the following internal steps:
a. Click on the object to select it.
Select Menu Bar => View => Tool Bars => Customize => Drawing; near the bottom is "Crop"; click on it; close the Customize window.
Use the "Crop" icon to grab the middle object handles; remove ("trim") that portion of the picture that is not the oscilloscope.

◊ 29. Size the remaining picture by grabbing the corner object handles; a double-ended arrow will appear; increase or decrease its size by moving in the direction of the arrow.

◊ 30. Move the location of the picture object using the keyboard "arrow" keys to a location of your preference; note the effect on the location of the text lines.

31. Print your final document if you so desire.

32. Close both the Word™ and the VEE Pro documents without saving them.

Lab 9.4 – Using VEE Pro to Prepare and Directly Print Reports in Microsoft Word™

This lab will show you how to load data into Microsoft Word™ and automatically print the resulting document.

Printing reports in Microsoft Word™

1. Open LAB9-3; immediately Save As … LAB9-4.
2. Change the Note Pad to read:
   This program is designed to immediately print the
   Test Results and its Graph in Word.
3. Move the Note Pad so it is beside and to the right of the largest Formula Object.
4. Select Menu Bar => Device => Formula; place it below and to the left of the largest Formula Object.
5. Select Menu Bar => Flow => If/Then/Else; place it to the right of the Formula Object in step 4; change its Properties box to read: Is Printer Configured?; change its input pin label from A to **str**.
6. Select Menu Bar => Device => Formula; place it to the left of Formula in step 4.
7. Select Menu Bar => Device => Formula; place it to the right of If/Then/Else.
8. Connect the four objects as shown in Figure 9.14; change the formula expressions as shown in Figure 9.14; you must delete terminals on some of the Formula objects.

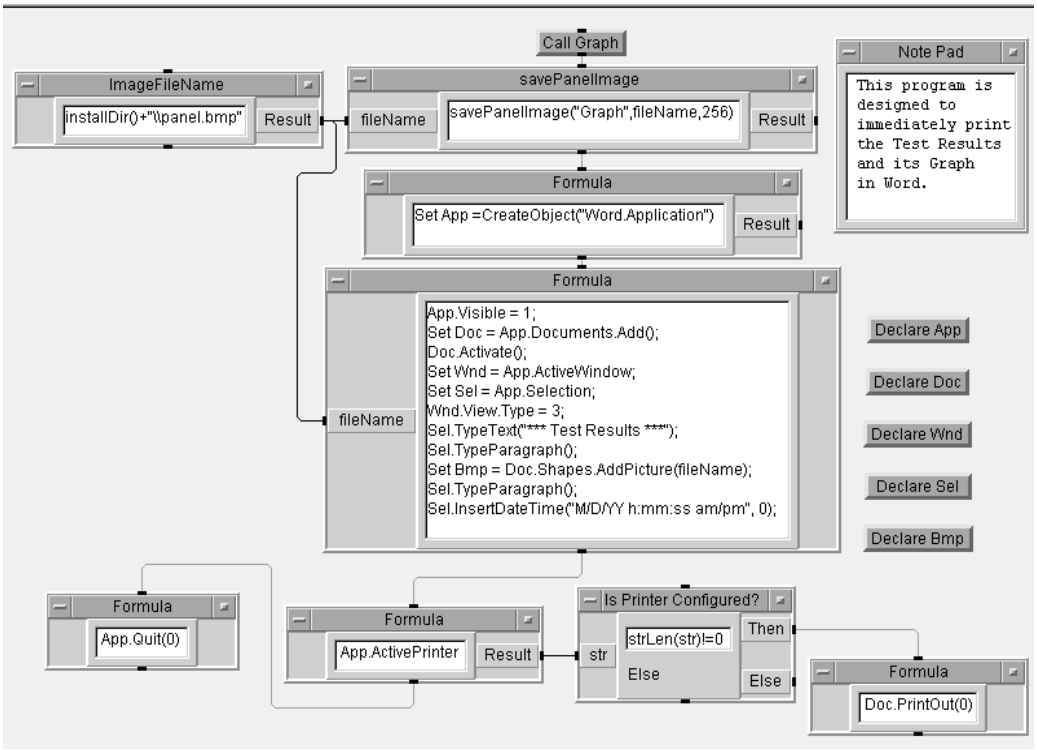

Figure 9.14. The complete Lab 9.4 program

9. Save this program again.
10. Run this program; the results should automatically print and the Word™ document will not remain on the screen.
    Note: It may be necessary to key your computer to select the desired printer and its configuration.
11. Cut the left-most Formula Object; re-run this program; the Word™ document will remain on the screen for you to modify the document as you so desire.
12. Save this program as LAB9-4a.
    Note: When closing LAB9-4a, you will be prompted: Do you want to save changes… because of the addition of the Word™ document during running. This provides you with the option of saving the Word™ document with a name of your preference.

Lesson 9 Summary

This lesson addressed using ActiveX to access Microsoft Excel™ in the preparation of spreadsheets. (VEE Pro allows you to document your test data and save it both electronically and as a hard copy.)
Excel™, accessed via ActiveX, is so flexible that you are limited in its applications to your everyday test documentation only by the amount of time that you can devote to developing new and meaningful presentations. It is imperative that you consider the potential reader of your reports and spreadsheets so that you can prepare them in a standard format.
Your time is precious; so is the time of your co-workers and managers.

Lab 9.1 showed you how to transfer VEE Pro data to Microsoft Excel™. Declared global variables and function calls are also introduced so you can learn to generate spreadsheets.

Lab 9.2 showed you how to create a generic template. It also allowed you to store test data as arrays and to modify the template to fill cells with data in any specified format.

Lab 9.3 showed you how to send text, a screen-dump of a VEE Pro pop-up panel with an XY display, and a time stamp to a Microsoft Word™ document.

Lab 9.4 showed you how to load data into Microsoft Word™ and automatically print the resulting document.

You are now ready to use VEE Pro to communicate more sophisticated data to Excel™. Also, see – under VEE Pro Help; Contents and Index – the Guide to Agilent VEE Example Programs. The examples are a subset under Microsoft Excel™. There are a variety of preprogrammed Excel™ spreadsheets.