

---

## How to use MS Excel to calculate data provided by COMPASS

### Summary

This article describes the setup requirements to use Microsoft Excel to perform calculations used by **COMPASS**. The information pertains to **COMPASS for Pressure** and **COMPASS for Flow**. A common use for Excel is within a Relationship Macro. Relationship Macros are designed to convert a raw output value (or values) into a Final output value. For example, a relationship macro can be used to determine a fully compensated pressure or flow value from a raw pressure, temperature and voltage. Many users may already have spreadsheets available that perform the specific math. So although the math required to determine the fully compensated value can be directly performed by the Relationship Macro, an existing solution using Excel already exists and is well understood by users. In this case calling the Excel solution is efficient and sustainable. During test execution, references to Excel are not visible to the user. Data is collected and reported by using Excel in the background.

In this article a Relationship Macro is used to write data to specific Excel fields and read back a calculated field value to **COMPASS**. **COMPASS** simply emulates a user typing in specific Excel fields and reading a value back. Therefore all operations normally handled by Excel are still available. Including all math and statistical functions, Excel Macros, links to other spreadsheets, database connectivity, etc. This mechanism can be implemented for any device used by **COMPASS**.

### Article Topics

- Device Setup
- Relationship Macros
- Excel Automation

### See Also

APN7320 (Report using MS Excel)

### Requirements

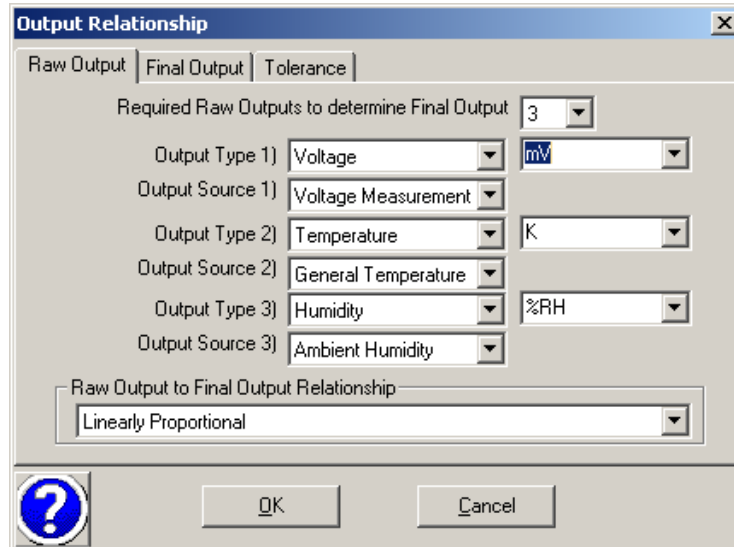
The following items are required to implement this setup.

- **COMPASS for Pressure Enhanced** or **COMPASS for Flow**
- Microsoft Excel 2002 or newer
- Knowledge on creating Excel spreadsheets

### Special Considerations

Prior to creating macros and devices that reference Excel, consider the source of data that will be provided to Excel. This determines the objects that must be referenced within the **COMPASS** macro and what outputs

must be included with the device setup. Outputs directly associated with a device (see figure below) are provided as parameters to the Relationship macro: *Out1*, *Out2* and *Out3*. If ambient conditions or other elements not specifically associated with the device are used, the data must be retrieved through the COMPASS object model: `cCOMPASS.cConfig.AmbPrs.FinalOutput`. When possible it is best to associate required outputs directly with a device. This helps to consolidate the custom functionality and simplifies unit conversion issues. In this example a combination of both schemes is used.



Outputs Directly Associated with a Device

## The Spreadsheet

In this example, the spreadsheet is used to determine an absolute pressure from a barometer and a voltage output pressure device that has temperature sensitivity. The only calculations are in cell B7. This is for simplicity of example and is in no way a requirement. As previously noted any and all desired Excel functionality can be used to determine the value provided to **COMPASS**.

Cells B2 through B5 have specific input values: device range, voltage output, ambient pressure and ambient temperature respectively. The **COMPASS** macro must update these fields with the corresponding data from **COMPASS** in order to determine the final pressure value in cell B7. After all input cells are updated, the value in cell B7 must be returned to **COMPASS** to represent the final output value.

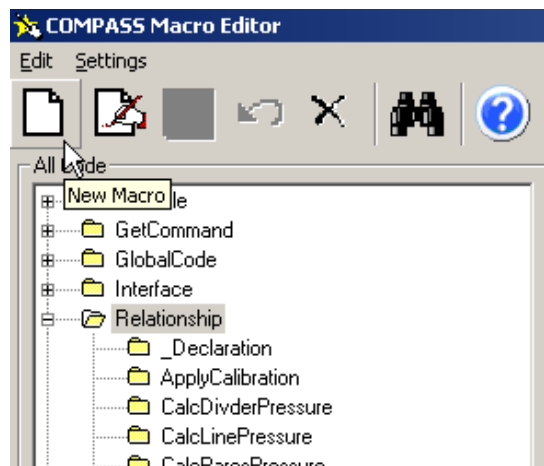
	A	B	C	D	E	F	G
1							
2	Range	100					
3	Voltage (V)	2					
4	Ambient Pressure (kPa)	98.452					
5	Ambient Temperature	20.22					
6							
7	Final Calc (kPa)	120.4056					
8							

Spreadsheet

## Macro Creation

Use the **[Tools][COMPASS Macro Editor]** to launch the Macro Editor. Two types of macros must be created to support this device: One macro to create and use the spreadsheet and a separate macro to remove the spreadsheet from memory when a test completes. The relationship macro is selected when creating the device in question, a DUT in this example. The macro to free the memory is selected as a Post Test Macro.

Select the *Relationship* category and press the **[New]** icon to create a new Relationship Macro (see Figure).



Create New Relationship Macro

A default macro with a random name is created. Change the name to something appropriate for the function of the macro. The code below details the macro. The first section of the macro is used to create the Excel Object and store a reference to the object within the **COMPASS** device. The second part of the macro is used to condition the outputs to be in the proper unit of measure as required by the spreadsheet example. There are no special requirements of the spreadsheet other than the data entered in a specific cell or cells triggers the calculation of the desired data in a different cell.

### Relationship Macro: CalculateExcelPressure

```

*****
'FOR HELP ON VBSCRIPT, PLEASE VISIT:
'http://www.webfl1.com/scriptdoc/vbscript/VBSTOC.htm
*****
'This Function must return the device output
'in the Final Output Unit by manipulating the
'parameters provided.
'
'Out1      :By default, this is the raw formatted, output of a
'           device. In other cases, it is final output of a
'           test configured device in the configured output unit.
'Out2      :Final output of a test configured device in the
'           configured output unit.
'Out3      :Final output of a test configured device in the
'           configured output unit.
'ParamID   :Parameter ID of the device .
'cCalc     :Calculation Class of the output.

'The value is returned by setting the function name =
'to the calculated value..
'
'For example: Relationship35778 = 1 / Out1^.5
*****
Function CalculateExcelPressure (Out1, Out2, Out3, ParamID, cCalc)

If cCalc.GetParamData(0) = 0 Then
    cDebug.LogStatus "Initializing Excel Object"
    'The Excel object has not been created
    'Create the object
    'Create Excel Object
    Set MyExcel = CreateObject("excel.application")
    MyExcel.DisplayAlerts = False

    'The path to the Excel File. Could be anywhere
    'or based on the device Model:
    'Model = cCalc.RangeMain.GetParent.Model
    'FileName = "C:\DHI\COMPASS for Pressure\" & Model & "\ExcelCalc.xls"
    FileName = "C:\DHI\COMPASS for Pressure\ExcelCalc.xlsx"

    'Open target spreadsheet
    Set MyWorkBook = MyExcel.Workbooks.Open(FileName)

```

Use debug statements to log data to the Macro Spy Window to simplify troubleshooting

Reference comments for additional detail

```

'Store the Excel Object and Workbook in COMPASS
Set cCalc.Obj1 = MyExcel
Set cCalc.Obj2 = MyWorkBook

'Set the flag that the device was initialized
cCalc.SetParamData 0, 1
cDebug.LogStatus "Excel Object Initialization complete"
End If

'Get the Excel Data into simple variables
V = Out1 'The first Raw Output is the voltage
cDebug.LogStatus "Voltage:" & V

P1 = cCOMPASS.cConfig.AmbPrs.FinalOutput
PUnit = cCOMPASS.cConfig.AmbPrs.RangeMain.UnitFinal
dType = cCOMPASS.cConfig.AmbPrs.RangeMain.TypeFinal 'always 0 for pressure devices

'Convert the output into kPa as required by the spreadsheet
'Reference Macro Editor Help for detail on the UnitConversion function
P = cCOMPASS.UnitConversion(CDbl(P1),3, CInt(pUnit),CInt(dType))
cDebug.LogStatus "Pressure:" & P

T1 = cCOMPASS.cConfig.AmbTemp.FinalOutput
TUnit = cCOMPASS.cConfig.AmbTemp.RangeMain.UnitFinal
dType = cCOMPASS.cConfig.AmbTemp.RangeMain.TypeFinal 'always 1 for temperature devices

'Convert the output into C as required by the spreadsheet
T = cCOMPASS.UnitConversion(CDbl(T1),0, CInt(TUnit),CInt(dType))
cDebug.LogStatus "Temperature:" & T

'Get the device full scale value in kPa
FS1 = cCalc.RangeMain.MaxFinal
PUnit = cCalc.RangeMain.UnitFinal
FS = cCOMPASS.UnitConversion(CDbl(FS1),3, CInt(PUnit), 0)
cDebug.LogStatus "FS:" & FS

'The Excell Workbook is stored in Obj2 of the
'calculation class. Get the relevant sheet from
'the workbook. Sheets can be referenced
'as text and as a number
'
'Set MySheet1 = cCalc.Obj2.Sheets("CalcSheet")
Set MySheet1 = cCalc.Obj2.Sheets(1)

'Write contents of cell
MySheet1.Range("B2").Value = FS
MySheet1.Range("B3").Value = V
MySheet1.Range("B4").Value = P
MySheet1.Range("B5").Value = T

' Read contents of cell
NewP = MySheet1.Range("B7").Value
cDebug.LogStatus "Excel Response:" & NewP

CalculateExcelPressure = NewP

End Function

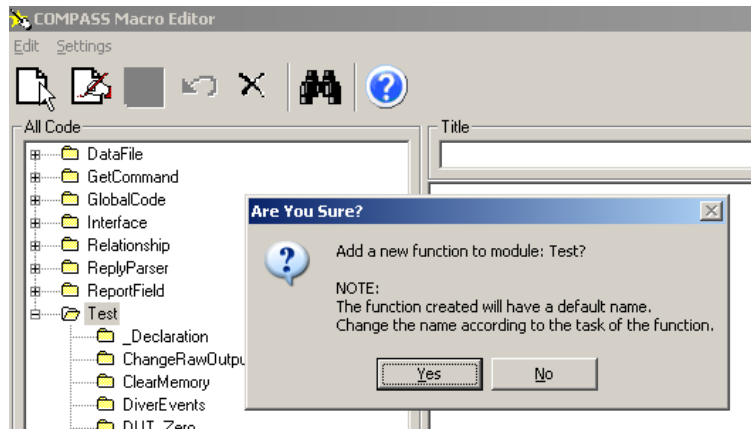
```

Reference the Help file for details on the functions and object constants.

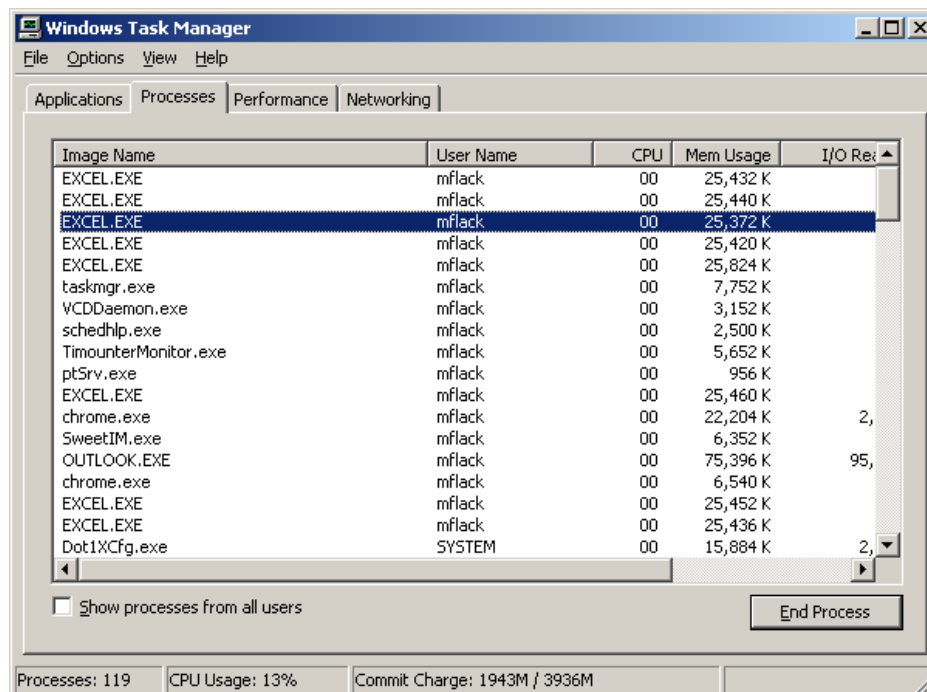
Type casting is required to force the proper data types in VBScript. **CDbl**, and **CInt** create a Double precision number and an Integer as required by the UnitConversion function.

Return value to **COMPASS** by setting desired value to the Function name.

Select the *Test* category and press the **[New]** icon to create a new Test Macro (see Figure). This macro will be used as a Post Test macro to free the memory used by **COMPASS**. If this macro is not used, multiple references to Excel will exist in the process view of Task Manager. The user must manually end each task or re-boot the PC to free up the memory used by each of these instances.



**Create New Test Macro**



**Task Manager when Excel memory is not cleared**

Similar to creating a new Relationship Macro, a Test Macro also begins with a default function.

```

Test Macro: ClearMemory

*****
'FOR HELP ON VBSCRIPT, PLEASE VISIT:
'http://www.webfl.com/scriptdoc/vbscript/VBSTOC.htm
*****
'Test Macro do not have a return value.
'Manipulate the test or device collection as desired.
'iT The current temperature point in the test
'iL The current line pressure point in the test.
'iC The current pressure cycle in the test.
'IP The current pressure point in the test.
'cTest The test class .
'cConfig Configuration class that holds all active devices.
'
*****
Function ClearMemory(iT, iL, iC, iP, cTest, cConfig)

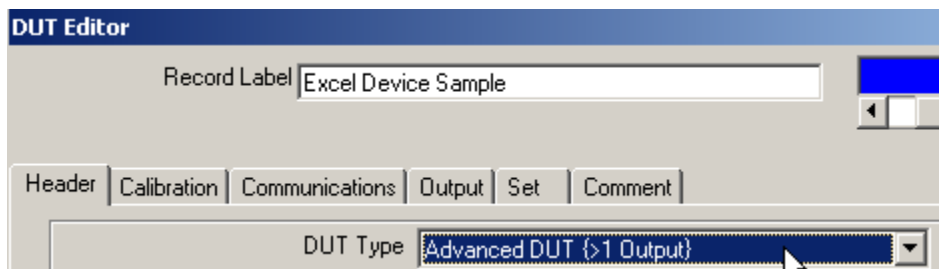
'Excel application is stored in Obj1 of DUT Calculation
'class. Quit the application then end all references to the
'object.
cConfig.DUTPrs(1).Obj1.Quit
Set cConfig.DUTPrs(1).Obj2 = Nothing 'Free the Workbook Memory
Set cConfig.DUTPrs(1).Obj1 = Nothing 'Free the Excel app Memory

End Function

```

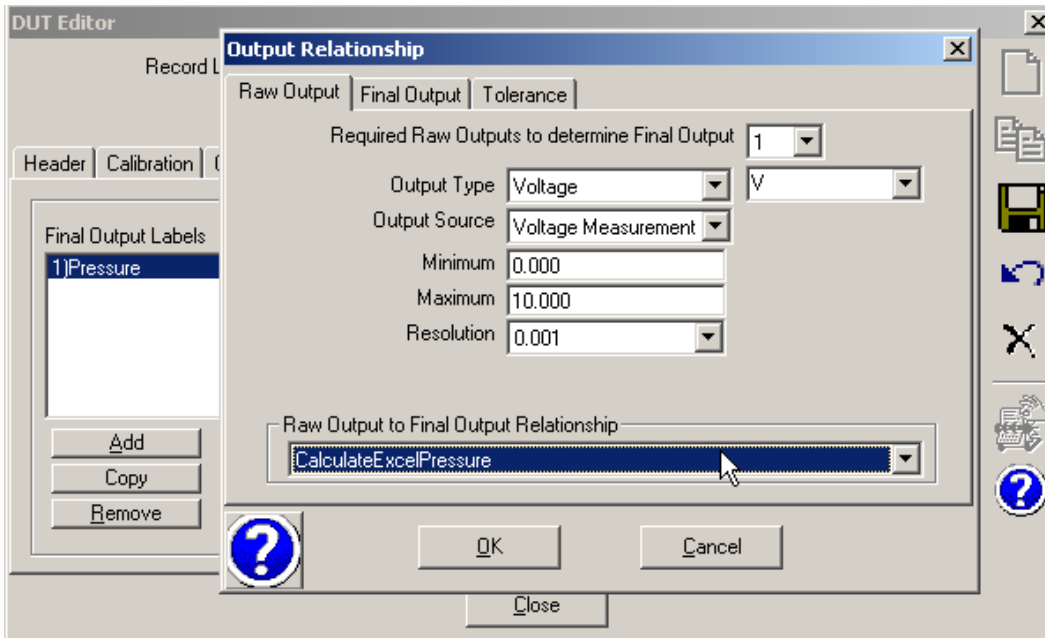
## Device setup

The setup of the device is consistent with all other **COMPASS** devices. The instrument can use remote commands over one of the supported interface selections or use the output of another instrument (a DMM) to determine the final pressure value. The only requirement is that the newly created Relationship macro listed above is associated with the device. Relationship Macros are only available in Advanced Device setups. Create a DUT (or Support Device), and make sure that the **<Advanced Device>** option is selected.

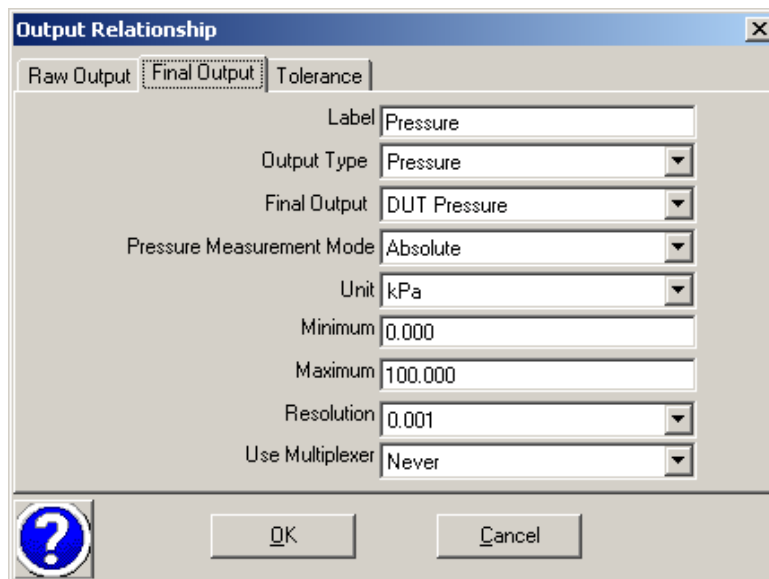


**Advanced Device selection is required**

On the **[Output]** create a new Output and select the **CalculateExcelPressure** Relationship (see Figure below). In this example a 0-10V Raw output will correspond to a 0 to 100 psi Final output. Save the device setup after the output is defined.



**Relationship Macro Selection**

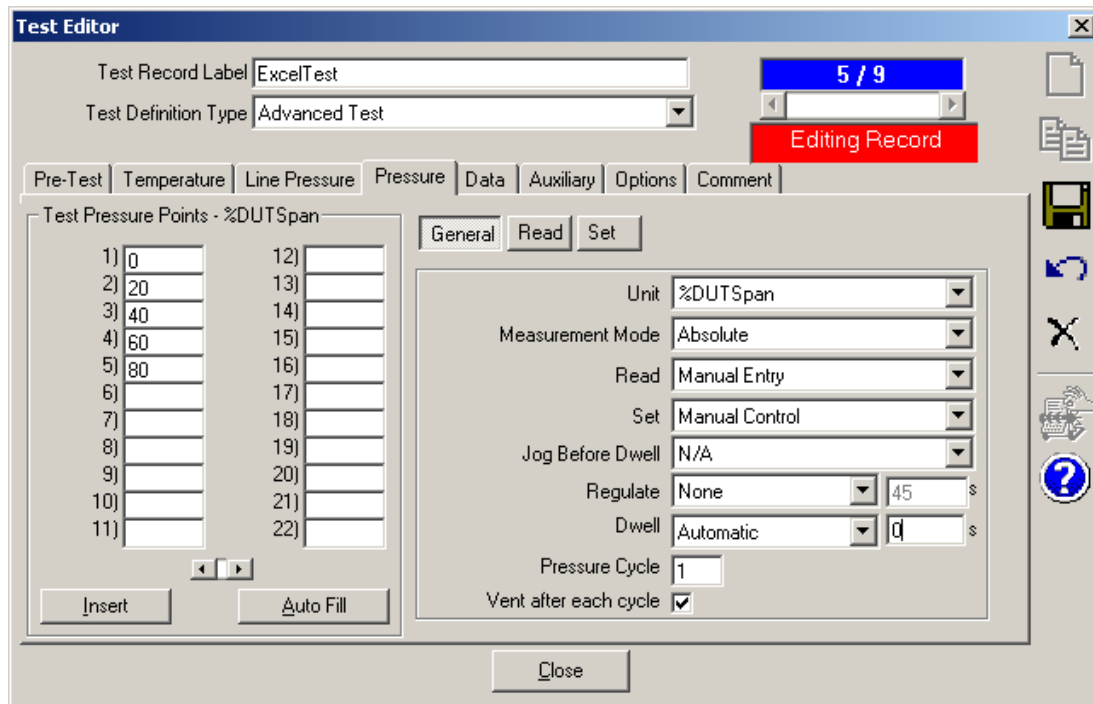


**Final Output Definition**



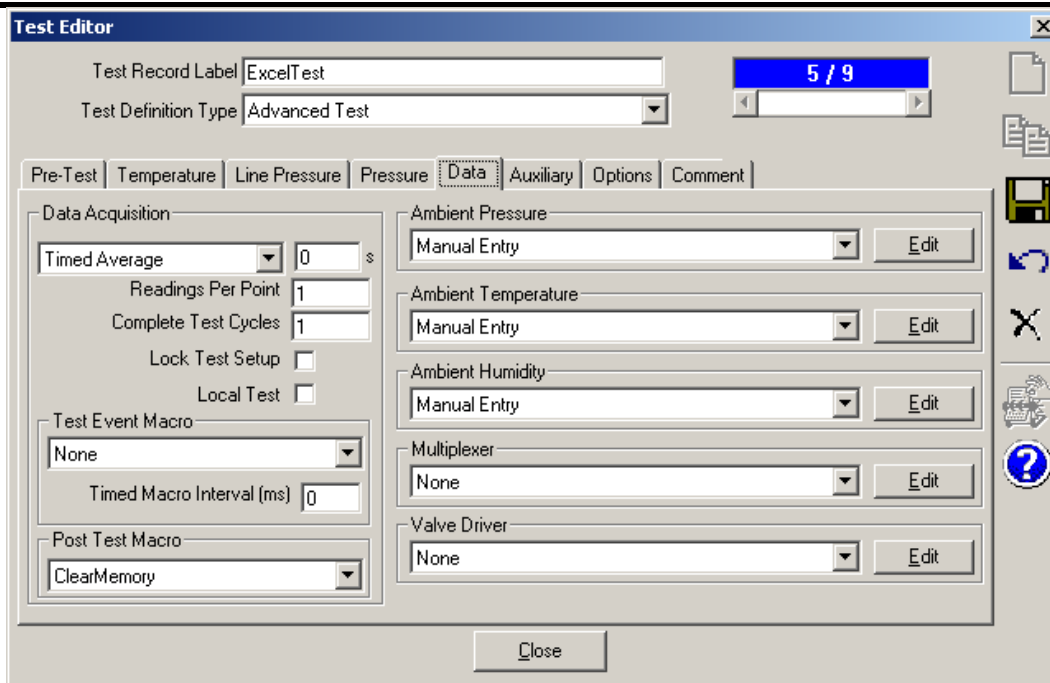
## Test Definition

Create a typical COMPASS test definition. The only requirement to support the Excel macro is to select the *ClearMemory* Post Test Macro. Select the **<Advanced Test>** test type to support macro selection. Define any series of pressure points (flow points in **COMPASS for Flow**). To simplify the troubleshooting process, create a test that uses manual reference, control and ambient condition devices. The test dwell and averaging time should be 0 as well. Choose the selections described in the figure below.



**Sample Test Pressure Tab selections**

On the **[Data]** tab, enter 0 as the averaging time and choose the *ClearMemory*, Post Test Macro. Duplicate the selections in the figure below.



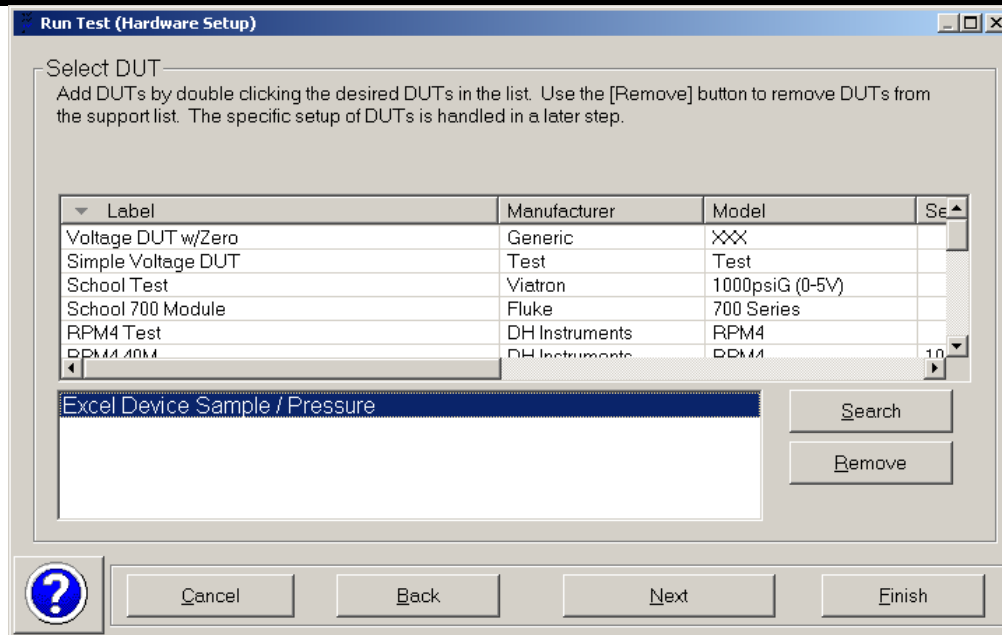
**Sample Test Data Tab Selections**

On the **[Options]** tab select the **<Execute post test macro on aborted tests.>** selection. This option ensures that the memory is released on all tests regardless of how the test ends.

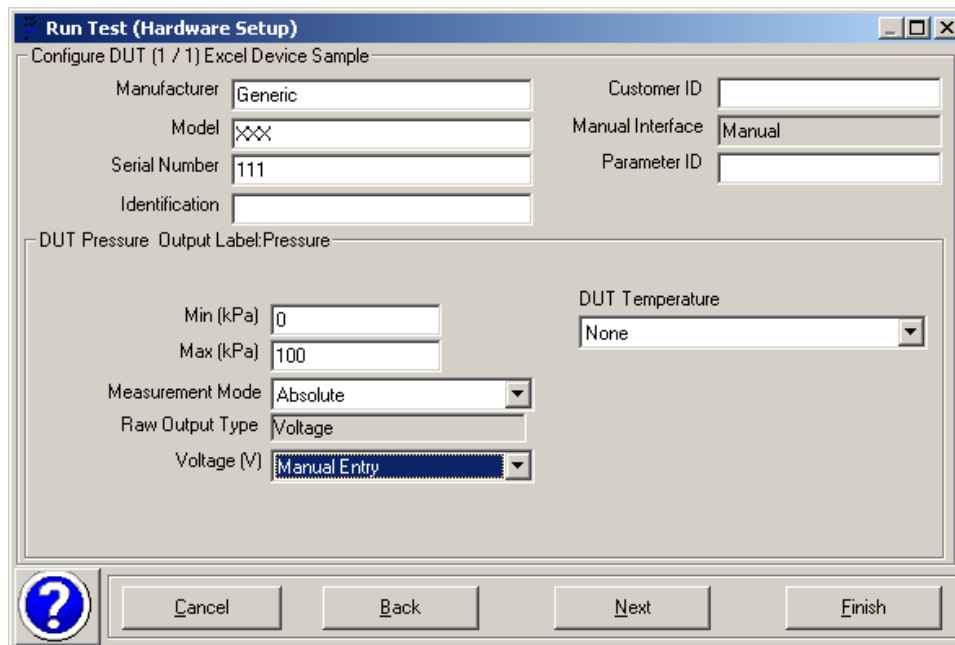
## Running a test and Troubleshooting

To verify the macro run the DUT and test created in this example. The manual nature of all of the other devices simplifies troubleshooting. After all operations of the macro are verified a regular test with non manual instruments can be used.

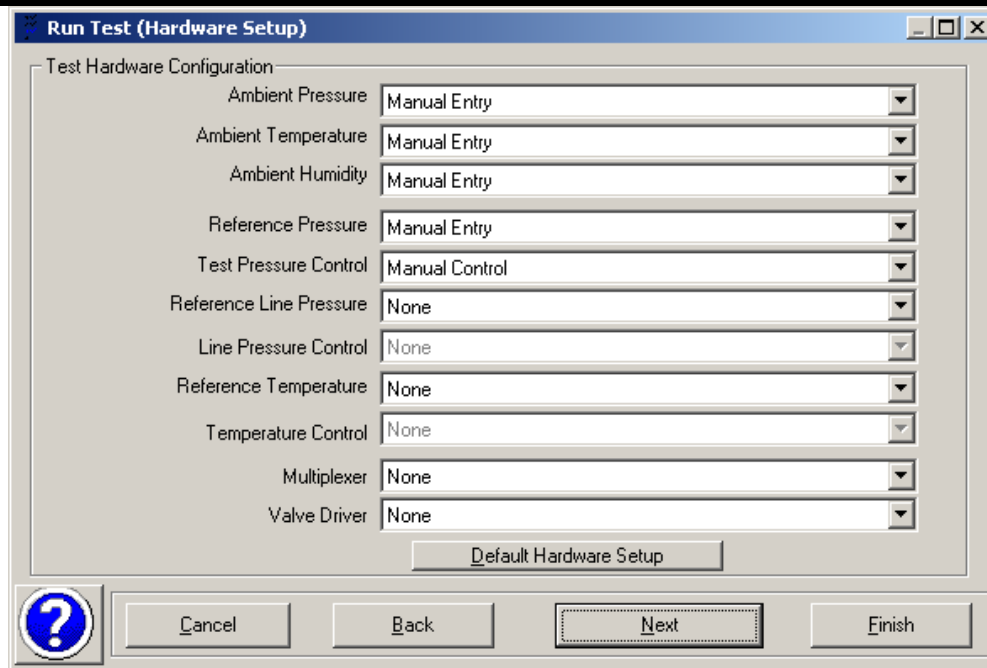
- Make sure that the spreadsheet is saved in the properly location defined by the path in the Relationship Macro. Close Excel to begin. It is acceptable to have Excel open when running a test that uses the Excel macro however it can complicate troubleshooting when the Macro has not previously been tested
- Use the **[Run][Run Test]** menu choice, select the DUT created in the Device Setup step of this example. Press **[Next]** to configure the device.



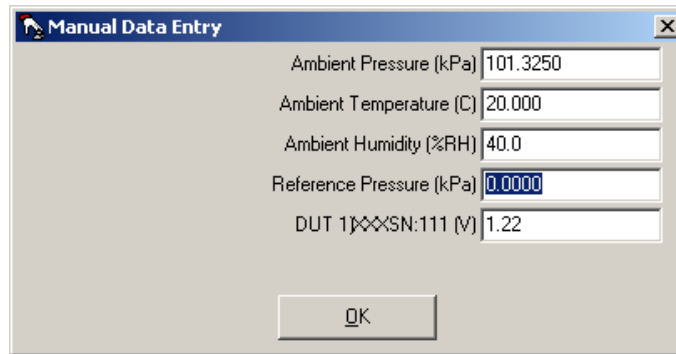
- Choose **<Manual Entry>** for the source of Voltage measurement and press **[Next]**.



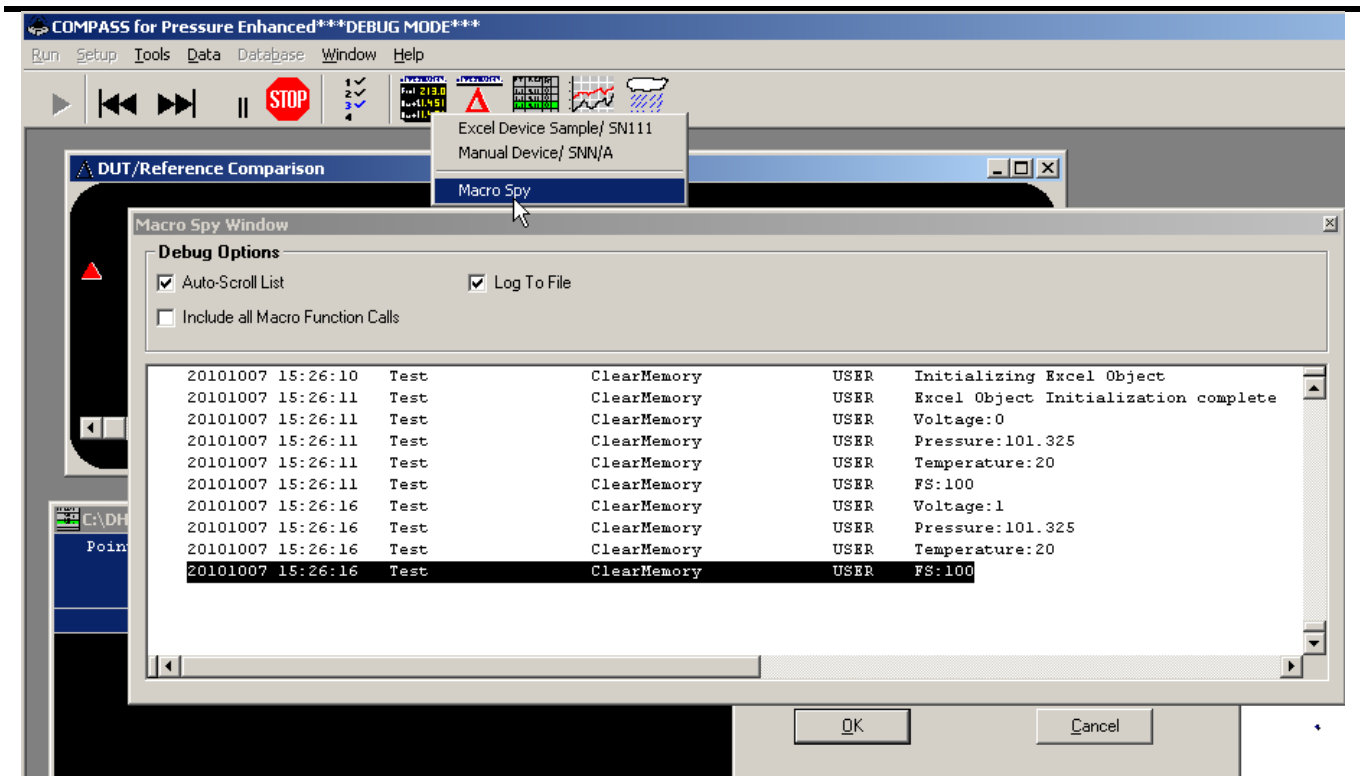
- Select the Excel Test previously created and press **[Next]**. Note that all test selections should be Manual Entry requiring no further selections. Press **[Finish]** to begin the test.



- The test begins with a prompt to set the target pressure. Press enter and let the dwell time complete. A manual entry data screen will display. Enter the relevant data and press enter.



- At this point the DUT pressure display should update with the information calculated by the Excel spreadsheet. If the data does not update, use the Macro Spy to troubleshoot the problem. The Macro Spy is accessed by using the selection at the bottom of the Run Screen button on the main program toolbar. The toolbar option to select the Macro Spy is only present after a macro runs. Therefore it will not be available at the beginning of the test when data has not been collected.



- All debug messages in the macro along with any errors will display in the Macro Spy. If there is an error, Pause **COMPASS** edit the macro, then Un-Pause and verify the proper functionality. Note the relationship macro only fires when input data relevant to the device changes. Since all instruments are manual, input data will change only when the manual entry data in the test is updated. Use the Back/Next buttons to repeat points as required until all macro related errors are complete. When running with live instruments, the macro is called each time one of the input instruments changes it's output.

When all operations are verified enhance the test and devices to use automated equipment as required.