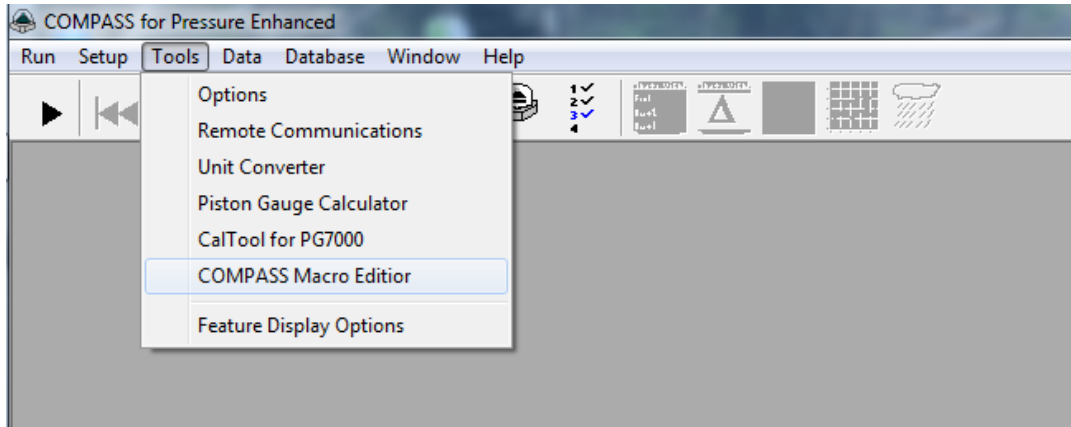


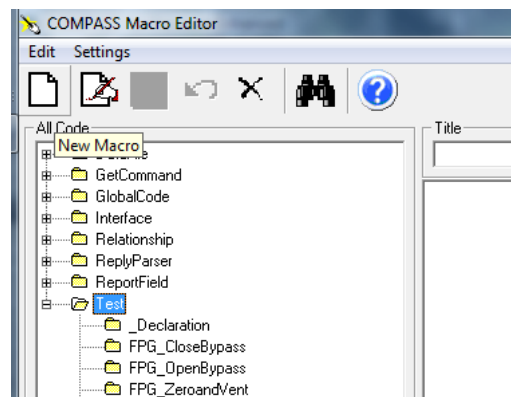
COMPASS for Pressure (or Flow)

Test macro to log data to a file with commands specified in the macro

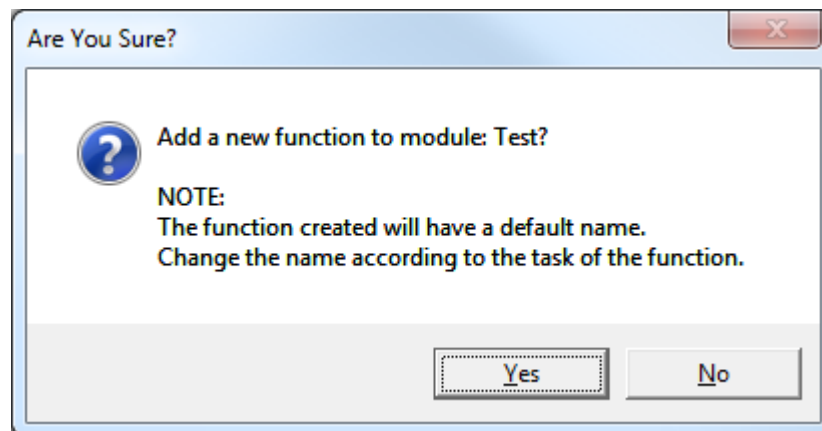
1. Open the COMPASS Macro Editor by this menu path; [Tools], <COMPASS Macro Editor>



2. Select "Test" then click the "New Macro" icon (blank white piece of paper) to start a new test macro.



3. Click [Yes] button to add a new function to module: Test

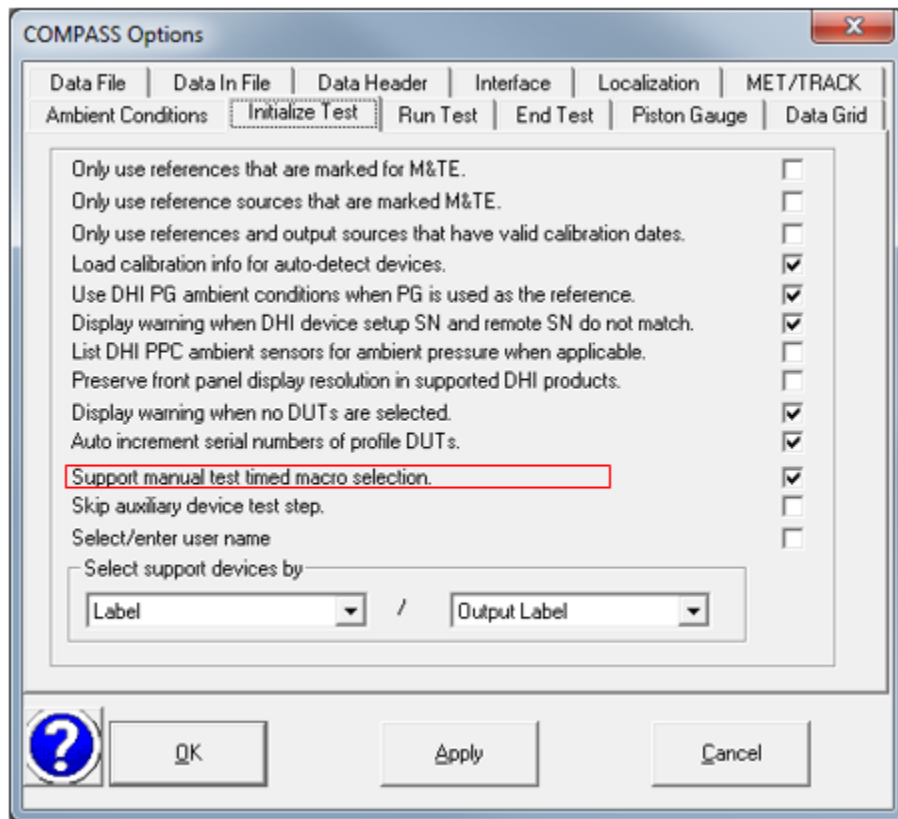


4. Select all of the text on the window and delete it
5. Copy and Paste all of the last two pages of this document into the window. Use [Ctrl]+[v] to paste as right-clicking does not work in macro editor

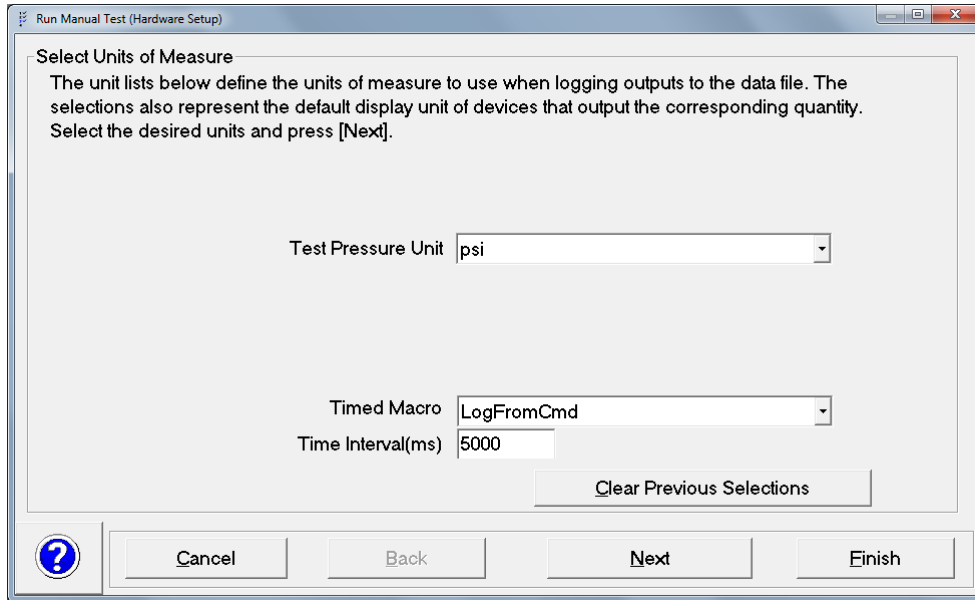
6. Add, delete or edit commands as necessary. Make “n” in “Max = n” equal the number of commands.

```
Max = 5
cmds (0) = "PPOS"
cmds (1) = "speed"
cmds (2) = "PR"
cmds (3) = "amb"
cmds (4) = "PPCPR"
```

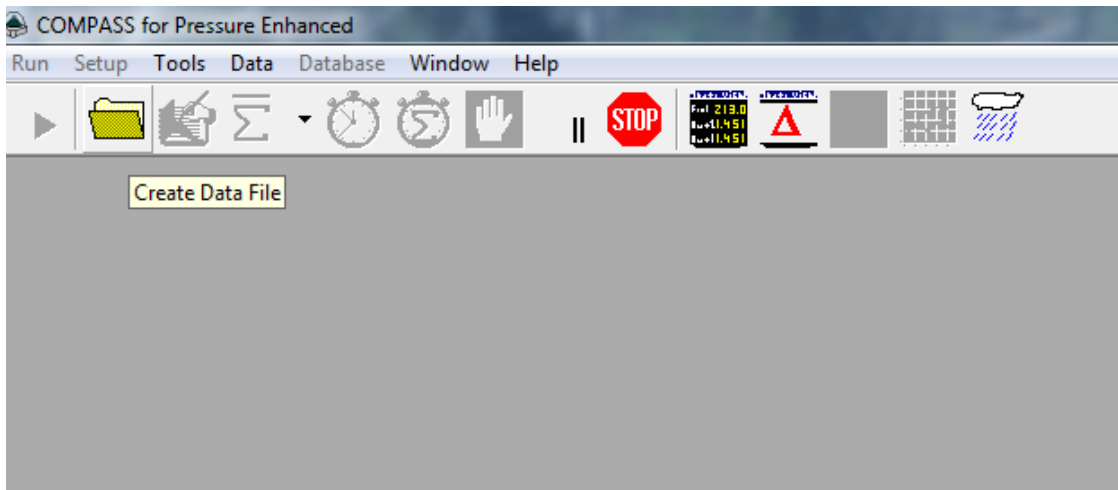
- 7. Enter “LogFromCmd” as the name in the “Title” box at the top and click the “Save” icon (disk icon). Error check will automatically take place, any errors will be shown and macro will be saved if no errors.
- 8. If you want to edit the commands in the future open the COMPASS Macro Editor and click the “Edit macro” icon. Save when done.
- 9. Close the COMPASS Macro Editor
- 10. If you want to be able to use the Test macro during a Manual Test you have to enable this. Open [Tools], <Options> window and select the <Initialize Test> tab
- 11. Make sure the “Support manual test timed macro selection” checkbox is checked then click [Apply], then [OK]



- To use in a manual test select the Test macro on the first Initialization screen. If the timer interval is zero (0) the macro will not run.



- Proceed through remainder of initialization screens like you normally do
- Click the "Create DataFile" icon (yellow folder), otherwise macro will not log data



- If the macro is working you will occasionally see the commands at the bottom left corner of the screen.



- Open the log file in Notepad or Excel and check that it is being populated with data. Location is shown on top line of the COMPASS data grid.

Point	Status	Set Point (psi)	Reference Pressure (psi)	DU1 Raw Out2 (Pa)
1	T		8.5070	
2	T		6.0740	
3	T		9.5554	
4	RT		574.2251	
5	RT		560.9169	

- Open the log file in Excel to analyze results. Use [Data], <Text to Table> to separate data into columns.

19. To use with a test definition the test must be an Advanced Test to have the “Test Event Macro” box in the “Data” tab. Choose the “LogFromCMD” macro in this box and select a timer interval in milli-seconds. If the timer interval is zero (0) the macro will not run.

The screenshot shows the 'Test Editor' window with the 'Data' tab selected. The window title is 'Test Editor' and it has a close button in the top right corner. The 'Test Record Label' is 'PG-AMH Test' and the 'Test Definition Type' is 'Advanced Test'. A blue status bar shows '16 / 29'. The 'Data' tab is active, showing various configuration options. On the left, the 'Data Acquisition' section includes 'Timed Average' (20 s), 'Readings Per Point' (1), 'Complete Test Cycles' (1), 'Lock Test Setup' (unchecked), and 'Local Test' (unchecked). The 'Test Event Macro' is set to 'LogFromCmd' with a 'Timed Macro Interval (ms)' of 5000. The 'Post Test Macro' is set to 'None'. On the right, there are five sensor configuration sections: 'Ambient Pressure' (Default), 'Ambient Temperature' (Default), 'Ambient Humidity' (Default), 'Multiplexer' (None), and 'Valve Driver' (None). Each section has an 'Edit' button. A 'Close' button is at the bottom center. A vertical toolbar on the right contains icons for file operations, a question mark, and other functions.

Section	Parameter	Value	Action
Data Acquisition	Timed Average	20 s	
	Readings Per Point	1	
	Complete Test Cycles	1	
	Lock Test Setup	<input type="checkbox"/>	
	Local Test	<input type="checkbox"/>	
Test Event Macro	Test Event Macro	LogFromCmd	
	Timed Macro Interval (ms)	5000	
Post Test Macro	Post Test Macro	None	
Sensors	Ambient Pressure	Default	Edit
	Ambient Temperature	Default	Edit
	Ambient Humidity	Default	Edit
	Multiplexer	None	Edit
	Valve Driver	None	Edit

```

'*****
'Test Macros 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
'*****

'Declare variable LogFileName
Dim LogFileName

Function LogFromCmd(iT, iL, iC, iP, cTest, cConfig)
Dim cmds(12)

'On Error Resume Next

'Change "Max =" value to number of commands you want to log
'and add/enter commands to log below
Max = 5
cmds(0) = "PPOS"
cmds(1) = "speed"
cmds(2) = "PR"
cmds(3) = "amb"
cmds(4) = "PPCPR"

Step = cCOMPASS.CurrentTestStep

''' COMPASS Run State (see Macro Editor help file)
''' 0 - Set for common operations, Run, Edit, Options, etc. The bit
'''      is cleared when the operation is complete (Data file Is created)
''' 1 - A COMPASS setup is being edited
''' 2 - Test initialization is active
''' 3 - Running a Test Definition
''' 4 - Running a Manual Test
''' 5 - Test initialization is complete and all instruments have been
'''      initialized. The value is set before a Pre-Test macro is executed
''' 6 - Data files have been created as part of a Test Definition
'''      or a Manual Test.

'Check if data file exists yet
If (cCOMPASS.COMPASSRunState And 2 ^ 6) = 0 Then
    'Data files have not even been created
    'cDebug.LogStatus "Run State: " & ccompass.COMPASSRunState
Exit Function

ElseIf LogFileName = "" Then
    'cDebug.LogStatus "LogFileName Before Naming: " & LogFileName
    'We need a filename for the log file, Use the DUT data file name
    'with a *.log extension using the same file name
temp = trim(cCOMPASS.DataCollection(1).FileName)
    'cDebug.LogStatus "FileName .dat: " & temp
LogFileName = left(temp, Len(temp)-3) & "log"
    'cDebug.LogStatus "LogFileName After Naming: " & LogFileName

```

```

'Create header line for log file and write to log file
For i = 0 To max
    sText = sText & cmds(i) & ";"
Next
rtv = SaveToTextFile(LogFileName,sText,True,";")
End If

sText = ""
For i = 0 To max
cCOMPASS.StatusDisplay "Reading: " & cmds(i)
If cCOMPASS.SystemAbort Then Exit Function
With cCOMPASS.cConfig.RefPrs(1).RangeMain
    reply = .ioSendCommand(CStr(cmds(i)), False, True)
End With
sText = sText & reply & ";"
Next

'write data to log file
LogFromCmd = SaveToTextFile(LogFileName,sText,True,";")

End Function

'////////////////////////////////////
'Open/Create a text file for Append and add the
'passed-in text to the file.  Optionally date/time
'stamp each entry, based on the boolean bLogDate
'parameter. Separator defines how the fields are
'separated, a ; is used by default.
'Close the file after each write, so it is available
'from multiple processes.
'////////////////////////////////////

Public Function SaveToTextFile(fName, sText, bLogDate, Separator)
Dim fso, f, sDate

Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.OpenTextFile(fName, 8, True)
    ''' 8 = open for append

If Separator = "" Then Separator = ";"
    ''' default separator to semi-colon

If bLogDate Then
    sDate = Date & Separator & formatDateTime(Time,3) & Separator
    ''' 3 = vbLongTime
    f.Write sDate & sText & vbcrLf
Else
    f.Write sText & vbcrLf
End If
f.Close

SaveToTextFile = True

End Function

```