Agilent E5070B/E5071B ENA Series RF Network Analyzers

Read or Write Trace Data

Second Edition

Agilent Technologies

No. 16000-95017

August 2002

Notices

The information contained in this document is subject to change without notice.

Agilent Technologies Japan, Ltd.

Component Test PGU-Kobe

1-3-2, Murotani, Nishi-ku, Kobe, Hyogo, 651-2241 Japan

Sample Program

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the customer’s internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purposes and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims any responsibility for the operation of the SAMPLE PROGRAMS to be uninterrupted or error-free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Read or Write Trace Data

Reading/Writing Measurement Data

This section describes how to process the E5070B/E5071B's internal data. You can use these internal data arrays: corrected data arrays, corrected memory arrays, formatted data arrays, formatted memory arrays, and stimulus data arrays. For more information on the internal data arrays, see Section "Internal Data Processing" in *E5070B/E5071B Programmer's Guide*.

To read/write a formatted data array or formatted memory array, use the following objects:

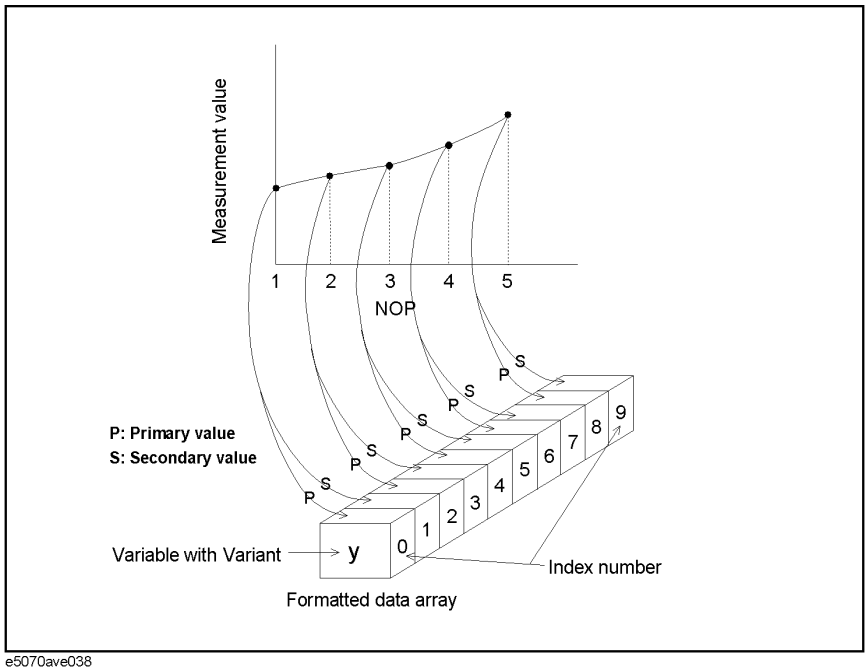
- **SCPI.CALCulate(Ch).SELected.DATA.FDATa**
- **SCPI.CALCulate(Ch).SELected.DATA.FMEMory**

To read a corrected data array, corrected memory array, or stimulus data array, use the following objects:

- **SCPI.CALCulate(Ch).SELected.DATA.SDATa**
- **SCPI.CALCulate(Ch).SELected.DATA.SMEMory**
- **SCPI.SENSe(Ch).FREQuency.DATA**

The E5070B/E5071B VBA allows you to deal with multiple pieces of data through variables of Variant type. Variant variables can contain any type of data, allowing you to deal with array data without being aware of the number of elements. For example, a formatted data array that includes 5 measurement points is stored as shown in Figure 1. Note that a formatted data array always contains 2 data items per measurement point, whichever data format is used. For more information on contained data, see Section "Internal Data Processing" in *E5070B/E5071B Programmer's Guide*; you can find a table that describes the relationship between contained data items and data formats.

Figure 1 **Example storing data into a Variant variable**



e5070ave038

NOTE

When you use one of the objects listed above, the base index number of the array is always 0 even if the declaration section contains the "Option Base 1" statement, which specifies the use of the base array index of 1.

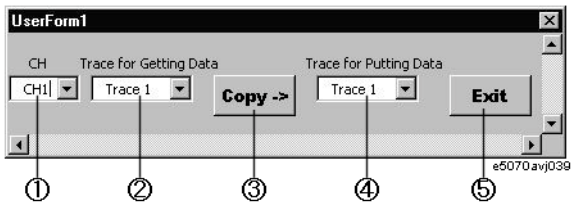
For example, you may wish to read the formatted data array for a particular trace in its entirety (including all measurement points), display the data in the echo window, and then write the data into another trace. How to implement such a process can be better understood with the aid of a sample program.

The sample program disk contains a sample program, named "read_write.vba", that demonstrates how to read and write measurement data. This VBA program consists of the following modules:

Object name	Module type	Content
frmReadWrite	UserForm	Reads, displays, and writes a formatted data array.
mdlReadWrite	Standard module	Invokes a UserForm.

When you run this VBA program, a window as shown in Figure 2 appears. For how to use each element in Figure 2, see the following description.

Figure 2 **The UserForm when running the Example 1 program**



1. The program lets the user specify the channel to be controlled.
2. The program lets the user specify which trace's formatted data array to read (source trace).
3. The program reads the formatted data array for the trace specified by the user, display the measurement results in the echo window, and write the data into the trace specified by the user. For detail, see the description of the code window.
4. The program lets the user specify which trace's formatted data array to overwrite (target trace).
5. The program exits, and the window disappears.

In Visual Basic Editor, open the UserForm (object name: frmReadWrite), and double-click the entire UserForm or the **Copy ->** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Copy** button on the UserForm (lines 10 to 520)

Lines 90 to 160	These lines identify the selected items in each list and store them into the variables TrGet, TrPut, and ActCh.
Lines 180 to 210	If the specified target trace is not displayed, these lines display that trace.
Lines 230 to 250	These lines make active the specified trace (TrGet: source trace) in the specified channel(ActCh) and hold the sweep.
Line 260	Reads the number of measurement points for the specified channel (ActCh) and stores that number into the Nop variable.
Line 280	Reads the formatted data array for the active trace (source trace) and store the data into the FmtData variable.
Line 290	Reads the stimulus array for the specified channel (ActCh) and stores the data into the Freq variable.
Line 330	Reads the data format for the active trace (source trace) and store it into the Fmt variable.
Lines 340 to 350	These lines display the echo window in the lower part of the LCD screen.
Lines 360 to 470	The lines display, in the echo window, each point along with one measured value (the odd part of the index is always 0) and a frequency if the Fmt is "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", or "UPH"; or along with two measured values and a frequency if Fmt$ returns any other string.
Line 490	Makes active the specified trace (TrPut: target trace) in the specified channel(ActCh).
Line 500	Writes the formatted data array (FmtData) into the active trace (target trace).

Procedure called when the user clicks the **Exit** button on the UserForm (lines 540 to 580)

Line 560	Unloads the UserForm from the memory, and terminates the program.

Procedure that initializes the UserForm (lines 600 to 1020)

Lines 620 to 1000	When the program is launched, these lines add each list item and set the default value for each list.

Example 1 **Reading/displaying/writing a formatted data array (read_write.frm)**

```
10|  Private Sub cmdCopy_Click()
20|
30|    Dim X As Integer, Y As Integer, Z As Integer, I As Integer
40|    Dim ActCh As Long, TrGet As Long, TrPut As Long
50|    Dim TrCont As Long, Nop As Long
60|    Dim FmtData As Variant, Freq As Variant
70|    Dim Fmt As String
80|
90|    X = cboCh.ListIndex
```

```
 100|    ActCh = X + 1
 110|
 120|    Y = cboGet.ListIndex
 130|    TrGet = Y + 1
 140|
 150|    Z = cboPut.ListIndex
 160|    TrPut = Z + 1
 170|
 180|    TrCont = SCPI.CALCulate(ActCh).PARameter.Count
 190|    If TrCont < TrPut Then
 200|      SCPI.CALCulate(ActCh).PARameter.Count = TrPut
 210|    End If
 220|
 230|    SCPI.CALCulate(ActCh).PARameter(TrGet).SELect
 240|    SCPI.INITiate(ActCh).CONTinuous = False
 250|    SCPI.ABORt
 260|    Nop = SCPI.SENSe(ActCh).SWEep.POINts
 270|
 280|    FmtData = SCPI.CALCulate(ActCh).SELected.Data.FDATa
 290|    Freq = SCPI.SENSe(ActCh).FREQuency.Data
 300|
 310|  '''Displays the formatted data
 320|
 330|    Fmt = SCPI.CALCulate(ActCh).SELected.Format
 340|    SCPI.DISPlay.TABLe.TYPE = "ECHO"
 350|    SCPI.DISPlay.TABLe.STATe = True
 360|    Select Case Fmt
 370|      Case "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL",
"IMAG", "UPH"
 380|          ECHO "Nop", "Frequency(GHz)", "Data"
 390|          For I = 0 To Nop - 1
 400|              ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I)
 410|          Next I
 420|      Case Else
 430|          ECHO "Nop", "Frequency(GHz)", "Data1", "Data2"
 440|          For I = 0 To Nop - 1
 450|            ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I),
FmtData(2 * I + 1)
 460|          Next I
 470|    End Select
 480|
 490|    SCPI.CALCulate(ActCh).PARameter(TrPut).SELect
 500|    SCPI.CALCulate(ActCh).SELected.Data.FDATa = FmtData
 510|
 520|  End Sub
 530|
 540|  Private Sub cmdExit_Click()
 550|
 560|    Unload Me
 570|
 580|  End Sub
 590|
 600|  Private Sub UserForm_Initialize()
 610|
 620|    With cboCh
 630|      .AddItem "CH1"
 640|      .AddItem "CH2"
 650|      .AddItem "CH3"
```

```
 660|      .AddItem "CH4"
 670|      .AddItem "CH5"
 680|      .AddItem "CH6"
 690|      .AddItem "CH7"
 700|      .AddItem "CH8"
 710|      .AddItem "CH9"
 720|    End With
 730|
 740|    With cboGet
 750|      .AddItem "Trace 1"
 760|      .AddItem "Trace 2"
 770|      .AddItem "Trace 3"
 780|      .AddItem "Trace 4"
 790|      .AddItem "Trace 5"
 800|      .AddItem "Trace 6"
 810|      .AddItem "Trace 7"
 820|      .AddItem "Trace 8"
 830|      .AddItem "Trace 9"
 840|    End With
 850|
 860|    With cboPut
 870|      .AddItem "Trace 1"
 880|      .AddItem "Trace 2"
 890|      .AddItem "Trace 3"
 900|      .AddItem "Trace 4"
 910|      .AddItem "Trace 5"
 920|      .AddItem "Trace 6"
 930|      .AddItem "Trace 7"
 940|      .AddItem "Trace 8"
 950|      .AddItem "Trace 9"
 960|    End With
 970|
 980|    cboCh.ListIndex = 0
 990|    cboGet.ListIndex = 0
1000|    cboPut.ListIndex = 0
1010|
1020|  End Sub
```