APPLICATION NOTE

E5071C to E5080B Code Migration

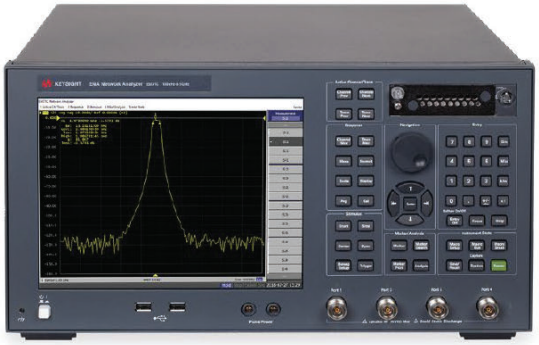
KEYSIGHT

Table of Contents

Overview

Switching from the E5071C to the E5080B simplifies complex measurements with integrated functionality and applications. You also get reliability and repeatability from optimal performance in dynamic range, trace noise, and frequency range.

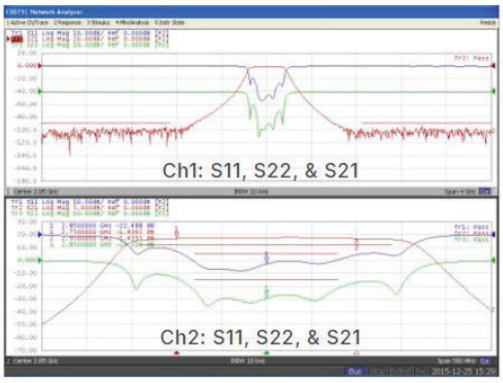
Upgrading an instrument in your workflow doesn't have to mean significant downtime to create SCPI commands all over again. The E5080's Code Emulation Mode enables you to keep using your existing E5071C code. This document describes how to use the E5080B's Code Emulation Mode as well as how to rewrite E5071C commands to run on the E5080B.



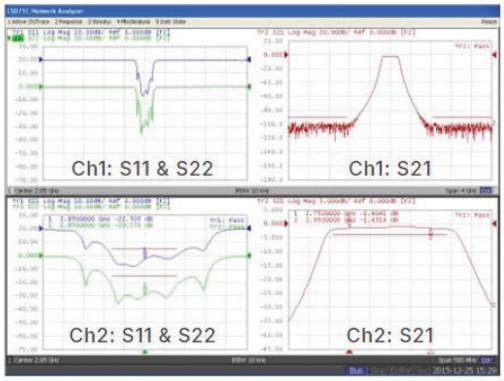
Part 1: Rewriting SCPI commands from E5071C to E5080B

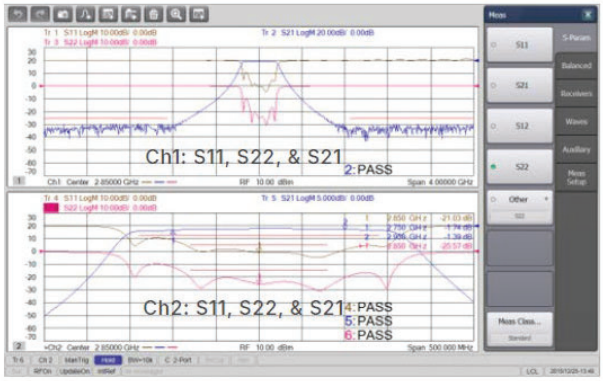
This chapter describes how to migrate the E5071C's remote control test program to the E5080B by rewriting the SCPI commands. The example program discussed here is a simple band pass filter test program. The overall flow of the example program is as follows;

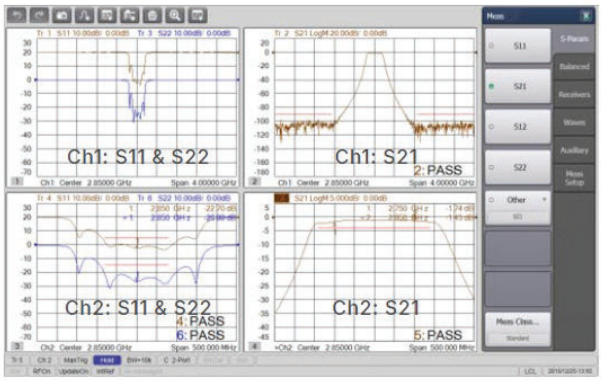
- Ch1: S11, S22, and S21 measurements in a wide span, linear sweep or segment sweep
- Ch2: S11, S22, and S21 measurements in a narrow span, linear sweep
- Plotting traces to two windows or four windows, as shown below.
- Setting limit lines and markers.
- Performing the 2-port full cal.
- Triggering the measurement and query the trace data.
- Saving and recalling state and user cal data.



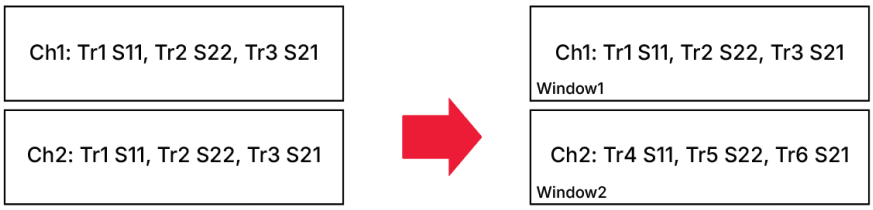
OR







Setting traces/channels/windows and measurement parameters - 2 windows



E5071C example

```
Ena.WriteString ":SYST:PRES", True

Ena.WriteString ":DISP:SPL D1_2", True

'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:PAR1:DEF S11", True
Ena.WriteString ":CALC1:PAR2:DEF S21", True
Ena.WriteString ":CALC1:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND1:SPL D1", True

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True

'Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:PAR1:DEF S11", True
Ena.WriteString ":CALC2:PAR2:DEF S21", True
Ena.WriteString ":CALC2:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND2:SPL D1", True

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True

'The following commands are also available.
'  Ena.WriteString ":CALC2:TRAC1:FORM MLOG", True
'  Ena.WriteString ":CALC2:TRAC2:FORM MLOG", True
'  Ena.WriteString ":CALC2:TRAC3:FORM MLOG", True
```

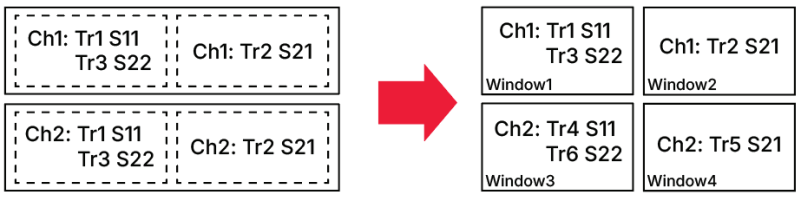
E5080B example

```
Ena.WriteString ":SYST:PRES", True

Ena.WriteString ":DISP:SPL 2", True

'Or the following commands can be used instead.
'  Ena.WriteString ":DISP:WIND1:STAT ON", True
'  Ena.WriteString ":DISP:WIND2:STAT ON", True
'  Ena.WriteString ":CALC1:PAR:DEL:ALL", True
'  Ena.WriteString ":CALC2:PAR:DEL:ALL", True

'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:MEAS1:PAR 'S11'", True
Ena.WriteString ":CALC1:MEAS2:PAR 'S21'", True
Ena.WriteString ":CALC1:MEAS3:PAR 'S22'", True

Ena.WriteString ":CALC1:MEAS1:FORM MLOG", True
Ena.WriteString ":CALC1:MEAS2:FORM MLOG", True
Ena.WriteString ":CALC1:MEAS3:FORM MLOG", True

' Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:MEAS4:PAR 'S11'", True
Ena.WriteString ":CALC2:MEAS5:PAR 'S21'", True
Ena.WriteString ":CALC2:MEAS6:PAR 'S22'", True

Ena.WriteString ":CALC2:MEAS4:FORM MLOG", True
Ena.WriteString ":CALC2:MEAS5:FORM MLOG", True
Ena.WriteString ":CALC2:MEAS6:FORM MLOG", True
```

NOTE:

Unlike the E5071C, the E5080B's trace number is a running number for all channels. The measurement numbers MEAS1 to MEAS6 in the above E5080B commands correspond to the trace numbers Tr1 to Tr6 shown in the display.

Setting traces/channels/windows and measurement parameters - 4 windows



E5071C example

```
Ena.WriteString ":SYST:PRES", True
Ena.WriteString ":DISP:SPL D1_2", True

'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:PAR1:DEF S11", True
Ena.WriteString ":CALC1:PAR2:DEF S21", True
Ena.WriteString ":CALC1:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND1:SPL D12", True

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True

'Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:PAR1:DEF S11", True
Ena.WriteString ":CALC2:PAR2:DEF S21", True
Ena.WriteString ":CALC2:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND2:SPL D12", True

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True

'The following commands are also available.
'  Ena.WriteString ":CALC2:TRAC1:FORM MLOG", True
'  Ena.WriteString ":CALC2:TRAC2:FORM MLOG", True
'  Ena.WriteString ":CALC2:TRAC3:FORM MLOG", True
```

NOTE:

TRAC<n> in the E5080B commands shown on the right corresponds not to Tr1 to Tr6 in the display but to the trace number in the specified window. For example, TRAC1 in WIND4 corresponds to Tr5 (=the string 'Meas_5') in this example.

E5080B example

```
Ena.WriteString ":SYST:PRES", True
Ena.WriteString ":DISP:SPL 4", True

'Or the following command can be used instead.
'  Ena.WriteString ":DISP:WIND1:STAT ON", True
'  Ena.WriteString ":DISP:WIND2:STAT ON", True
'  Ena.WriteString ":DISP:WIND3:STAT ON", True
'  Ena.WriteString ":DISP:WIND4:STAT ON", True
'  Ena.WriteString ":CALC1:PAR:DEL:ALL", True
'  Ena.WriteString ":CALC2:PAR:DEL:ALL", True

' Ch-1 & 2 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:DEF:EXT 'Meas_1','S11'", True
Ena.WriteString ":CALC1:PAR:DEF:EXT 'Meas_2','S21'", True
Ena.WriteString ":CALC1:PAR:DEF:EXT 'Meas_3','S22'", True
Ena.WriteString ":CALC2:PAR:DEF:EXT 'Meas_4','S11'", True
Ena.WriteString ":CALC2:PAR:DEF:EXT 'Meas_5','S21'", True
Ena.WriteString ":CALC2:PAR:DEF:EXT 'Meas_6','S22'", True

Ena.WriteString ":DISP:WIND1:TRAC1:FEED 'Meas_1'", True
Ena.WriteString ":DISP:WIND1:TRAC2:FEED 'Meas_3'", True
Ena.WriteString ":DISP:WIND2:TRAC1:FEED 'Meas_2'", True
Ena.WriteString ":DISP:WIND3:TRAC1:FEED 'Meas_4'", True
Ena.WriteString ":DISP:WIND3:TRAC2:FEED 'Meas_6'", True
Ena.WriteString ":DISP:WIND4:TRAC1:FEED 'Meas_5'", True

Ena.WriteString ":CALC1:MEAS1:FORM MLOG", True
Ena.WriteString ":CALC1:MEAS2:FORM MLOG", True
Ena.WriteString ":CALC1:MEAS3:FORM MLOG", True
Ena.WriteString ":CALC2:MEAS4:FORM MLOG", True
Ena.WriteString ":CALC2:MEAS5:FORM MLOG", True
Ena.WriteString ":CALC2:MEAS6:FORM MLOG", True

'Or the following commands can be used instead.
'  Ena.WriteString ":CALC1:PAR:SEL 'Meas_1'", True
'  Ena.WriteString ":CALC1:FORM MLOG", True
'  Ena.WriteString ":CALC1:PAR:SEL 'Meas_2'", True
'  Ena.WriteString ":CALC1:FORM MLOG", True
'  Ena.WriteString ":CALC1:PAR:SEL 'Meas_3'", True
'  Ena.WriteString ":CALC1:FORM MLOG", True
'  Ena.WriteString ":CALC2:PAR:SEL 'Meas_4'", True
'  Ena.WriteString ":CALC2:FORM MLOG", True
'  Ena.WriteString ":CALC2:PAR:SEL 'Meas_5'", True
'  Ena.WriteString ":CALC2:FORM MLOG", True
'  Ena.WriteString ":CALC2:PAR:SEL 'Meas_6'", True
'  Ena.WriteString ":CALC2:FORM MLOG", True
```

Y-axis scale setup

The following examples set RLEV (Y-axis reference level), RPOS (Y-axis reference position), and PDIV (dB/div) values of measurement traces.

E5071C example

```
'Ch-1 Y-axis scale setup
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:PDIV 20", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:PDIV 10", True

'Ch-2 Y-axis scale setup
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:PDIV 5", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:PDIV 10", True
```

E5080B example, for two window setup

```
'Window-1 Y-axis scale setup
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:PDIV 20", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND1:TRAC3:Y:PDIV 10", True

'Window-2 Y-axis scale setup
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND2:TRAC2:Y:PDIV 5", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND2:TRAC3:Y:PDIV 10", True
```

E5080B example, for four window setup

```
'Window-1 & 2 Y-axis scale setup
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND1:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND1:TRAC2:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND2:TRAC1:Y:PDIV 20", True

'Window-3 & 4 Y-axis scale setup
Ena.WriteString ":DISP:WIND3:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND3:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND3:TRAC1:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND3:TRAC2:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND3:TRAC2:Y:RPOS 7", True
Ena.WriteString ":DISP:WIND3:TRAC2:Y:PDIV 10", True
Ena.WriteString ":DISP:WIND4:TRAC1:Y:RLEV 0", True
Ena.WriteString ":DISP:WIND4:TRAC1:Y:RPOS 9", True
Ena.WriteString ":DISP:WIND4:TRAC1:Y:PDIV 5", True
```

Stimulus setup - Ch1 & 2: linear sweep

The following examples set the X-axis stimulus conditions as follows;

Ch1: Linear sweep, Center=2.85 GHz, Span=4 GHz, NOP=801, IFBW=10 kHz, Power=10 dBm
Ch2: Linear sweep, Center=2.85 GHz, Span=500 MHz, NOP=201, IFBW=10 kHz, Power=10 dBm

E5071C example

```
'Ch-1 linear sweep setup
Ena.WriteString ":SENS1:SWE:TYPE LIN", True
Ena.WriteString ":SENS1:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS1:FREQ:SPAN 4E9", True
Ena.WriteString ":SENS1:SWE:POIN 801", True
Ena.WriteString ":SENS1:BWID 10E3", True
Ena.WriteString ":SOUR1:POW 10", True

'Ch-2 linear sweep setup
Ena.WriteString ":SENS2:SWE:TYPE LIN", True
Ena.WriteString ":SENS2:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS2:FREQ:SPAN 500E6", True
Ena.WriteString ":SENS2:SWE:POIN 201", True
Ena.WriteString ":SENS2:BWID 10E3", True
Ena.WriteString ":SOUR2:POW 10", True
```

E5080B example

```
'Ch-1 linear sweep setup
Ena.WriteString ":SENS1:SWE:TYPE LIN", True
Ena.WriteString ":SENS1:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS1:FREQ:SPAN 4E9", True
Ena.WriteString ":SENS1:SWE:POIN 801", True
Ena.WriteString ":SENS1:BWID 10E3", True
Ena.WriteString ":SOUR1:POW 10", True

'Ch-2 linear sweep setup
Ena.WriteString ":SENS2:SWE:TYPE LIN", True
Ena.WriteString ":SENS2:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS2:FREQ:SPAN 500E6", True
Ena.WriteString ":SENS2:SWE:POIN 201", True
Ena.WriteString ":SENS2:BWID 10E3", True
Ena.WriteString ":SOUR2:POW 10", True
```

Stimulus setup - Ch1: segment sweep & Ch2: linear sweep

The following examples make the following segment table for the channel-1;

Segment-1: 850 MHz to 2.3 GHz, NOP=201, IFBW=10 kHz
Segment-2: 2.3 GHz to 3.4 GHz, NOP=201, IFBW=70 kHz
Segment-3: 3.4 GHz to 4.85 GHz, NOP=201, IFBW=10 kHz

E5071C example

```
Dim SegmData_ch1(18) As Variant

'Ch-1 segment sweep setup
SegmData_ch1(0) = 5
SegmData_ch1(1) = 0
SegmData_ch1(2) = 1
SegmData_ch1(3) = 0
SegmData_ch1(4) = 0
SegmData_ch1(5) = 0
SegmData_ch1(6) = 3
SegmData_ch1(7) = 850000000#
SegmData_ch1(8) = 2300000000#
SegmData_ch1(9) = 201
SegmData_ch1(10) = 10000#
SegmData_ch1(11) = 2300000000#
SegmData_ch1(12) = 3400000000#
SegmData_ch1(13) = 201
SegmData_ch1(14) = 70000#
SegmData_ch1(15) = 3400000000#
SegmData_ch1(16) = 4850000000#
SegmData_ch1(17) = 201
SegmData_ch1(18) = 10000#

Ena.WriteString ":FORM:DATA ASC", True
Ena.WriteString ":SENS1:SWE:TYPE SEGM", True
Ena.WriteString ":DISP:WIND1:X:SPAC LIN", True
Ena.WriteString ":SENS1:SEGM:DATA ", False
Ena.WriteList SegmData_ch1, ASCIIType_R8, ",", True

Ena.WriteString ":SOUR1:POW 10", True


' Ch-2 linear sweep setup
Ena.WriteString ":SENS2:SWE:TYPE LIN", True
Ena.WriteString ":SENS2:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS2:FREQ:SPAN 500E6", True
Ena.WriteString ":SENS2:SWE:POIN 201", True
Ena.WriteString ":SENS2:BWID 10E3", True
Ena.WriteString ":SOUR2:POW 10", True
```

E5080B example

```
Dim SegmData_ch1(11) As Variant

'Ch-1 linear sweep setup
SegmData_ch1(0) = 850000000#
SegmData_ch1(1) = 2300000000#
SegmData_ch1(2) = 201
SegmData_ch1(3) = 10000#
SegmData_ch1(4) = 2300000000#
SegmData_ch1(5) = 3400000000#
SegmData_ch1(6) = 201
SegmData_ch1(7) = 70000#
SegmData_ch1(8) = 3400000000#
SegmData_ch1(9) = 4850000000#
SegmData_ch1(10) = 201
SegmData_ch1(11) = 10000#

Ena.WriteString ":SENS1:SEGM:DEL:ALL", True

Ena.WriteString ":SENS1:SEGM1:ADD", True
Ena.WriteString ":SENS1:SEGM1:STAT ON", True
Ena.WriteString ":SENS1:SEGM1:FREQ:STAR " & SegmData_ch1(0), True
Ena.WriteString ":SENS1:SEGM1:FREQ:STOP " & SegmData_ch1(1), True
Ena.WriteString ":SENS1:SEGM1:SWE:POIN " & SegmData_ch1(2), True
Ena.WriteString ":SENS1:SEGM1:BWID:CONT ON", True
Ena.WriteString ":SENS1:SEGM1:BWID " & SegmData_ch1(3), True
Ena.WriteString ":SENS1:SEGM2:ADD", True
Ena.WriteString ":SENS1:SEGM2:STAT ON", True
Ena.WriteString ":SENS1:SEGM2:FREQ:STAR " & SegmData_ch1(4), True
Ena.WriteString ":SENS1:SEGM2:FREQ:STOP " & SegmData_ch1(5), True
Ena.WriteString ":SENS1:SEGM2:SWE:POIN " & SegmData_ch1(6), True
Ena.WriteString ":SENS1:SEGM2:BWID:CONT ON", True
Ena.WriteString ":SENS1:SEGM2:BWID " & SegmData_ch1(7), True
Ena.WriteString ":SENS1:SEGM3:ADD", True
Ena.WriteString ":SENS1:SEGM3:STAT ON", True
Ena.WriteString ":SENS1:SEGM3:FREQ:STAR " & SegmData_ch1(8), True
Ena.WriteString ":SENS1:SEGM3:FREQ:STOP " & SegmData_ch1(9), True
Ena.WriteString ":SENS1:SEGM3:SWE:POIN " & SegmData_ch1(10), True
Ena.WriteString ":SENS1:SEGM3:BWID:CONT ON", True
Ena.WriteString ":SENS1:SEGM3:BWID " & SegmData_ch1(11), True

Ena.WriteString ":SENS1:SWE:TYPE SEGM", True
Ena.WriteString "SENS:SEGM:X:SPAC LIN", True

Ena.WriteString ":SOUR1:POW 10", True

' Ch-2 linear sweep setup
Ena.WriteString ":SENS2:SWE:TYPE LIN", True
Ena.WriteString ":SENS2:FREQ:CENT 2.85E9", True
Ena.WriteString ":SENS2:FREQ:SPAN 500E6", True
Ena.WriteString ":SENS2:SWE:POIN 201", True
Ena.WriteString ":SENS2:BWID 10E3", True
Ena.WriteString ":SOUR2:POW 10", TrueEna.WriteString ":SENS2:BWID 10E3", True
Ena.WriteString ":SOUR2:POW 10", True
```

Limit line setup

The following examples set the limit lines as follows;

Ch1 S21 measurement: -90 dB max limit at 850 MHz to 2 GHz, -90 dB max limit at 3.7 GHz to 4.85 GHz
Ch2 S11 measurement: -15 dB max limit at 2.77 to 2.93 GHz
Ch2 S21 measurement: -4 dB min limit at 2.71 to 2.99 GHz
Ch2 S22 measurement: -15 dB max limit at 2.77 to 2.93 GHz

E5071C example

```
Dim LimitData_ch1_tr2(10), LimitData_ch2_tr1(5),
LimitData_ch2_tr2(5), LimitData_ch2_tr3(5) As Variant

Ena.WriteString ":FORM:DATA ASC", True

' Ch-1 trace-2 limit line setup
LimitData_ch1_tr2(0) = 2                    '# of limit lines
LimitData_ch1_tr2(1) = 1                    '1:max, 2:min
LimitData_ch1_tr2(2) = 850000000#           'X-value at start point
LimitData_ch1_tr2(3) = 2000000000#          'X-value at end point
LimitData_ch1_tr2(4) = -90                  'Y-value at start point
LimitData_ch1_tr2(5) = -90                  'Y-value at end point
LimitData_ch1_tr2(6) = 1                    '1:max, 2:min
LimitData_ch1_tr2(7) = 3700000000#          'X-value at start point
LimitData_ch1_tr2(8) = 4850000000#          'X-value at end point
LimitData_ch1_tr2(9) = -90                  'Y-value at start point
LimitData_ch1_tr2(10) = -90                 'Y-value at end point

Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:LIM:DATA ", False
Ena.WriteList LimitData_ch1_tr2, ASCIIType_R8, ",", True
Ena.WriteString ":CALC1:SEL:LIM:DISP:STAT ON", True
Ena.WriteString ":CALC1:SEL:LIM:STAT ON", True

' Ch-2 trace-1 limit line setup
LimitData_ch2_tr1(0) = 1
LimitData_ch2_tr1(1) = 1
LimitData_ch2_tr1(2) = 2770000000#
LimitData_ch2_tr1(3) = 2930000000#
LimitData_ch2_tr1(4) = -15
LimitData_ch2_tr1(5) = -15

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:LIM:DATA ", False
Ena.WriteList LimitData_ch2_tr1, ASCIIType_R8, ",", True
Ena.WriteString ":CALC2:SEL:LIM:DISP:STAT ON", True
Ena.WriteString ":CALC2:SEL:LIM:STAT ON", True

' Ch-2 trace-2 limit line setup
LimitData_ch2_tr2(0) = 1
LimitData_ch2_tr2(1) = 2
LimitData_ch2_tr2(2) = 2710000000#
LimitData_ch2_tr2(3) = 2990000000#
LimitData_ch2_tr2(4) = -4
LimitData_ch2_tr2(5) = -4

Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:LIM:DATA ", False
Ena.WriteList LimitData_ch2_tr2, ASCIIType_R8, ",", True
Ena.WriteString ":CALC2:SEL:LIM:DISP:STAT ON", True
Ena.WriteString ":CALC2:SEL:LIM:STAT ON", True

' Ch-2 trace-3 limit line setup
LimitData_ch2_tr3(0) = 1
LimitData_ch2_tr3(1) = 1
LimitData_ch2_tr3(2) = 2770000000#
LimitData_ch2_tr3(3) = 2930000000#
LimitData_ch2_tr3(4) = -15
LimitData_ch2_tr3(5) = -15

Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:LIM:DATA ", False
Ena.WriteList LimitData_ch2_tr3, ASCIIType_R8, ",", True
Ena.WriteString ":CALC2:SEL:L IM:DISP:STAT ON", True
Ena.WriteString ":CALC2:SEL:LIM:STAT ON", True

'The following commands are also available.
' Ena.WriteString ":CALC2:TRAC3:LIM:DATA ", False
' Ena.WriteList LimitData_ch2_tr3, ASCIIType_R8, ",", True
' Ena.WriteString ":CALC2:TRAC3:LIM:DISP:STAT ON", True
' Ena.WriteString ":CALC2:TRAC3:LIM:STAT ON", True
```

E5080B example

```
Dim LimitData_meas2(7), LimitData_meas4(4), LimitData_meas5(4), LimitData_meas6(4) As Variant

‘ Ch-1 trace-2 limit line setup
LimitData_meas2(0) = 850000000#
LimitData_meas2(1) = 2000000000#
LimitData_meas2(2) = -90
LimitData_meas2(3) = -90
LimitData_meas2(4) = 3700000000#
LimitData_meas2(5) = 4850000000#
LimitData_meas2(6) = -90
LimitData_meas2(7) = -90

Ena.WriteString ":CALC1:MEAS2:LIM:SEGM1:STIM:STAR " & LimitData_meas2(0), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM1:STIM:STOP " & LimitData_meas2(1), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM1:AMPL:STAR " & LimitData_meas2(2), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM1:AMPL:STOP " & LimitData_meas2(3), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM1:TYPE LMAX", True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM2:STIM:STAR " & LimitData_meas2(4), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM2:STIM:STOP " & LimitData_meas2(5), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM2:AMPL:STAR " & LimitData_meas2(6), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM2:AMPL:STOP " & LimitData_meas2(7), True
Ena.WriteString ":CALC1:MEAS2:LIM:SEGM2:TYPE LMAX", True
Ena.WriteString ":CALC1:MEAS2:LIM:STAT ON", True

‘ Ch-2 trace-1 limit line setup
LimitData_meas4(0) = 2770000000#
LimitData_meas4(1) = 2930000000#
LimitData_meas4(2) = -15
LimitData_meas4(3) = -15

Ena.WriteString ":CALC2:MEAS4:LIM:SEGM1:STIM:STAR " & LimitData_meas4(0), True
Ena.WriteString ":CALC2:MEAS4:LIM:SEGM1:STIM:STOP " & LimitData_meas4(1), True
Ena.WriteString ":CALC2:MEAS4:LIM:SEGM1:AMPL:STAR " & LimitData_meas4(2), True
Ena.WriteString ":CALC2:MEAS4:LIM:SEGM1:AMPL:STOP " & LimitData_meas4(3), True
Ena.WriteString ":CALC2:MEAS4:LIM:SEGM1:TYPE LMAX", True
Ena.WriteString ":CALC2:MEAS4:LIM:STAT ON", True

‘ Ch-2 trace-2 limit line setup
LimitData_meas5(0) = 2710000000#
LimitData_meas5(1) = 2990000000#
LimitData_meas5(2) = -4
LimitData_meas5(3) = -4

Ena.WriteString ":CALC2:MEAS5:LIM:SEGM1:STIM:STAR " & LimitData_meas5(0), True
Ena.WriteString ":CALC2:MEAS5:LIM:SEGM1:STIM:STOP " & LimitData_meas5(1), True
Ena.WriteString ":CALC2:MEAS5:LIM:SEGM1:AMPL:STAR " & LimitData_meas5(2), True
Ena.WriteString ":CALC2:MEAS5:LIM:SEGM1:AMPL:STOP " & LimitData_meas5(3), True
Ena.WriteString ":CALC2:MEAS5:LIM:SEGM1:TYPE LMIN", True
Ena.WriteString ":CALC2:MEAS5:LIM:STAT ON", True
```

```
' Ch-2 trace-3 limit line setup
LimitData_meas6(0) = 2770000000#
LimitData_meas6(1) = 2930000000#
LimitData_meas6(2) = -15
LimitData_meas6(3) = -15

Ena.WriteString ":CALC2:MEAS6:LIM:SEGM1:STIM:STAR " & LimitData_meas6(0), True
Ena.WriteString ":CALC2:MEAS6:LIM:SEGM1:STIM:STOP " & LimitData_meas6(1), True
Ena.WriteString ":CALC2:MEAS6:LIM:SEGM1:AMPL:STAR " & LimitData_meas6(2), True
Ena.WriteString ":CALC2:MEAS6:LIM:SEGM1:AMPL:STOP " & LimitData_meas6(3), True
Ena.WriteString ":CALC2:MEAS6:LIM:SEGM1:TYPE LMAX", True
Ena.WriteString ":CALC2:MEAS6:LIM:STAT ON", True

'Or the following commands can be used instead of the above commands if 'Meas_x' has been defined before.
'    Ena.WriteString ":CALC2:PAR:SEL 'Meas_6'", True
'    Ena.WriteString ":CALC2:LIM:SEGM1:STIM:STAR " & LimitData_meas6(0), True
'    Ena.WriteString ":CALC2:LIM:SEGM1:STIM:STOP " & LimitData_meas6(1), True
'    Ena.WriteString ":CALC2:LIM:SEGM1:AMPL:STAR " & LimitData_meas6(2), True
'    Ena.WriteString ":CALC2:LIM:SEGM1:AMPL:STOP " & LimitData_meas6(3), True
'    Ena.WriteString ":CALC2:LIM:SEGM1:TYPE LMAX", True
'    Ena.WriteString ":CALC2:LIM:STAT ON", True
```

Marker setup

The following examples put the markers as follows, in the MKR coupling OFF mode;

Ch2 S11 measurement: MKR1 at 2.85 GHz
Ch2 S21 measurement: MKR1 at 2.75 GHz, MKR2 at 2.95 GHz
Ch2 S22 measurement: MKR1 at 2.85 GHz

E5071C example

```
Ena.WriteString ":CALC2:MARK:COUP OFF", True
Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MARK1:X 2.85E9", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MARK1:X 2.75E9", True
Ena.WriteString ":CALC2:MARK2:STAT ON", True
Ena.WriteString ":CALC2:MARK2:X 2.95E9", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MARK1:X 2.85E9", True

'The following commands are also available.
'   Ena.WriteString ":CALC2:MARK:COUP OFF", True
'   Ena.WriteString ":CALC2:TRAC1:MARK1:STAT ON", True
'   Ena.WriteString ":CALC2:TRAC1:MARK1:X 2.85E9", True
'   Ena.WriteString ":CALC2:TRAC2:MARK1:STAT ON", True
'   Ena.WriteString ":CALC2:TRAC2:MARK1:X 2.75E9", True
'   Ena.WriteString ":CALC2:TRAC2:MARK2:STAT ON", True
'   Ena.WriteString ":CALC2:TRAC2:MARK2:X 2.95E9", True
'   Ena.WriteString ":CALC2:TRAC3:MARK1:STAT ON", True
'   Ena.WriteString ":CALC2:TRAC3:MARK1:X 2.85E9", True

Ena.WriteString ":DISP:WIND2:ANN:MARK:SING:STAT OFF", True
```

E5080B example

```
Ena.WriteString ":CALC2:MEAS4:MARK:COUP OFF", True

Ena.WriteString ":CALC2:MEAS4:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MEAS4:MARK1:X 2.85E9", True

Ena.WriteString ":CALC2:MEAS5:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MEAS5:MARK1:X 2.75E9", True
Ena.WriteString ":CALC2:MEAS5:MARK2:STAT ON", True
Ena.WriteString ":CALC2:MEAS5:MARK2:X 2.95E9", True

Ena.WriteString ":CALC2:MEAS6:MARK1:STAT ON", True
Ena.WriteString ":CALC2:MEAS6:MARK1:X 2.85E9", True

'Or the following commands can be used if 'Meas_x' has
'been defined before.
'   Ena.WriteString ":CALC2:PAR:SEL 'Meas_6'", True
'   Ena.WriteString ":CALC2:MARK1:STAT ON", True
'   Ena.WriteString ":CALC2:MARK1:X 2.85E9", True
```

NOTE:
The commands for querying the marker values are as follows;

```
Y-axis value (2-dimension data)
   E5071C   :CALC<ch>:TRAC<tnum>:MARK<mk>:Y?, or  :CALCulate<ch>:SEL:MARKer<mk>:Y?
   E5080B   :CALC<ch>:MEAS<mnum>:MARK<mk>:Y?

X-axis value
   E5071C   :CALC<ch>:TRAC<tnum>:MARK<mk>:X?, or  :CALCulate<ch>:SEL:MARKer<mk>:X?
   E5080B   :CALC<ch>:MEAS<mnum>:MARK<mk>:X?

Both X & Y-axis values (3-dimension data)
   E5071C   :CALC<ch>:TRAC<tnum>:MARK<mk>:DATA?, or  :CALCulate<ch>:SEL:MARKer<mk>:DATA?
   E5080B   N/A
```

Mechanical full 2-port cal

The following example performs the full 2-port (SOLT) cal using the E5071C's commands. The calibration is performed for the channel-1 and 2 together in the same connections as follows;

Port-1 open cal for Ch1 & 2 → Port-1 short cal for Ch1 & 2 → ... → Thru cal for Ch1 & 2 → Done.

The 85033D/E 3.5 mm mechanical cal kit is selected in this example.

E5071C example

```
Dim Dummy As Long

Ena.WriteString ":SENS1:CORR:COLL:METH:SOLT2 1,2", True
Ena.WriteString ":SENS2:CORR:COLL:METH:SOLT2 1,2", True

Ena.WriteString ":SENS1:CORR:COLL:CKIT 1", True   '85033E
Ena.WriteString ":SENS2:CORR:COLL:CKIT 1", True

MsgBox ("Connect OPEN to port-1")
Ena.WriteString ":SENS1:CORR:COLL:OPEN 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:OPEN 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-1")
Ena.WriteString ":SENS1:CORR:COLL:SHOR 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:SHOR 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-1")
Ena.WriteString ":SENS1:CORR:COLL:LOAD 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:LOAD 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect OPEN to port-2")
Ena.WriteString ":SENS1:CORR:COLL:OPEN 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:OPEN 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-2")
Ena.WriteString ":SENS1:CORR:COLL:SHOR 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:SHOR 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-2")
Ena.WriteString ":SENS1:CORR:COLL:LOAD 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:LOAD 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect THRU between port-1 & 2")
Ena.WriteString ":SENS1:CORR:COLL:THRU 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS1:CORR:COLL:THRU 2,1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:THRU 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:THRU 2,1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString ":SENS1:CORR:COLL:SAVE", True
Ena.WriteString ":SENS2:CORR:COLL:SAVE", True
```

The following example performs the same full 2-port cal using the E5080B's guided cal commands. The connector type (of the cal kit and the DUT) and the cal kit's model number must be specified with the string data for the analyzer ports to be used. With the unmodified default cal kit definition, the E5080B automatically performs the unknown thru cal (SOLR cal) for the test ports of the same connector gender like this example.

E5080B example (A)

```
Dim Dummy As Long

Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:INIT", True

Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:INIT", True

MsgBox ("Connect OPEN to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN3", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN3", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect OPEN to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN4", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN4", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN5", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN5", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN6", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN6", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect THRU betwen port-1 & 2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN7", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN7", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString "SENS1:CORR:COLL:GUID:SAVE", True
Ena.WriteString "SENS2:CORR:COLL:GUID:SAVE", True
```

This is another example routine of the full 2-port cal utilizing the prompt string messages like “Connect 3.5 MM FEMALE OPEN to port-1” provided by the E5080B’s guided cal.

E5080B example (B)

```
Dim numSteps As Integer, strPrompt As String
Dim i As Integer

   ‘You can confirm the names of supported cal kits as follows.
   ‘    Dim Connector, Calkit As String
   ‘    Ena.WriteString “SENS1:CORR:COLL:GUID:CONN:CAT?”
   ‘    Connector = Ena.ReadString
   ‘    MsgBox Connector
   ‘    Ena.WriteString “SENS1:CORR:COLL:GUID:CKIT:CAT? ‘APC 3.5 female’”
   ‘    Calkit = Ena.ReadString
   ‘    MsgBox Calkit

Ena.WriteString “:SENS1:CORR:COLL:GUID:CONN:PORT1 ‘APC 3.5 female’”, True
Ena.WriteString “:SENS1:CORR:COLL:GUID:CONN:PORT2 ‘APC 3.5 female’”, True
Ena.WriteString “:SENS1:CORR:COLL:GUID:CKIT:PORT1 ‘85033D/E’”, True
Ena.WriteString “:SENS1:CORR:COLL:GUID:CKIT:PORT2 ‘85033D/E’”, True
Ena.WriteString “:SENS1:CORR:COLL:GUID:INIT”, True

Ena.WriteString “:SENS2:CORR:COLL:GUID:CONN:PORT1 ‘APC 3.5 female’”, True
Ena.WriteString “:SENS2:CORR:COLL:GUID:CONN:PORT2 ‘APC 3.5 female’”, True
Ena.WriteString “:SENS2:CORR:COLL:GUID:CKIT:PORT1 ‘85033D/E’”, True
Ena.WriteString “:SENS2:CORR:COLL:GUID:CKIT:PORT2 ‘85033D/E’”, True
Ena.WriteString “:SENS2:CORR:COLL:GUID:INIT”, True

Ena.WriteString “:SENS1:CORR:COLL:GUID:STEPS?”, True
numSteps = Ena.ReadNumber

For i = 1 To numSteps
   Ena.WriteString “SENS1:CORR:COLL:GUID:DESC? “ + CStr(i), True
   strPrompt = Ena.ReadString
   MsgBox strPrompt
   Ena.WriteString “SENS1:CORR:COLL:GUID:ACQ STAN” + CStr(i), True
   Ena.WriteString “SENS2:CORR:COLL:GUID:ACQ STAN” + CStr(i), True
Next

Ena.WriteString “SENS1:CORR:COLL:GUID:SAVE”, True
Ena.WriteString “SENS2:CORR:COLL:GUID:SAVE”, True
```

2-port ECal

The following example performs the 2-port ECal for the channel-1 and 2 on the E5071C. The calibration is performed with a defined thru. To perform the unknown thru ECal, the command ":SENS:CORR:COLL:ECAL:UTHRU:STAT ON" should be added.

E5071C example

```
Dim Dummy As Long

' Add the following commands to perform the unknown thru cal.
'   Ena.WriteString ":SENS1:CORR:COLL:ECAL:UTHRU:STAT ON", True
'   Ena.WriteString ":SENS2:CORR:COLL:ECAL:UTHRU:STAT ON", True

MsgBox ("Connect ECal between port-1 & 2")

Ena.WriteString ":SENS1:CORR:COLL:ECAL:SOLT2 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString ":SENS2:CORR:COLL:ECAL:SOLT2 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

In the E5080B firmware rev.A.11.70 or above, the ECal can be easily performed by executing the single command ":SENS:CORR:COLL:GUID:ECAL:ACQ". The parameter "SOLR" specifies the unknown thru cal. The ECal kit selection command ":SENS:CORR:COLL:GUID:ECAL:SEL" can be omitted if you connect just one ECal module to the E5080B and use its factory default cal data, which normally appears at the top of the ECal kit list.

E5080B example (A)

```
Dim Dummy As Long

'   Ena.WriteString ":SENS1:CORR:COLL:GUID:ECAL:SEL 'N4431B ECal 03605'", True
'   Ena.WriteString ":SENS2:CORR:COLL:GUID:ECAL:SEL 'N4431B ECal 03605'", True

MsgBox ("Connect ECal between port-1 & 2")

Ena.WriteString ":SENS1:CORR:COLL:GUID:ECAL:ACQ SOLR,1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString ":SENS2:CORR:COLL:GUID:ECAL:ACQ SOLR,1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

This is another example routine of performing the 2-port ECal on the E5080B by using the conventional guided cal commands. It is necessary to specify the model and serial number of the ECal module, like 'N4431B ECal 03605'. You can verify these numbers on the E5080B front panel after connecting the ECal module, by pressing [Cal] and {ECal} keys. Or, you can query it with the SENS:CORR:COLL:GUID:CKIT:CAT? command.

E5080B example (B)

```
Dim Dummy As Long
Dim i, numSteps As Integer

Ena.WriteString "SENS1:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString "SENS1:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
 ' Ena.WriteString "SENS1:CORR:COLL:GUID:CONN:PORT3 'Not used'"
 ' Ena.WriteString "SENS1:CORR:COLL:GUID:CONN:PORT4 'Not used'"

Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:PORT1 'N4431B ECal 03605'", True
Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:PORT2 'N4431B ECal 03605'", True

    'Or you can query the name of the connected ECal module and set the test ports as follows.
    '    Dim Calkit As Variant
    '    Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:CAT? 'APC 3.5 female'"
    '    Calkit = Ena.ReadList
    '    Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:PORT1 '" & Calkit(1) & "'", True
    '    Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:PORT2 '" & Calkit(1) & "'", True

MsgBox ("Connect ECal between port-1 & 2")

Ena.WriteString "SENS1:CORR:COLL:GUID:INIT", True
Ena.WriteString "SENS1:CORR:COLL:GUID:STEPS?", True
numSteps = Ena.ReadNumber

For i = 1 To numSteps
   Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN" & CStr(i), True
Next

Ena.WriteString "SENS1:CORR:COLL:GUID:SAVE", True

Ena.WriteString "SENS2:CORR:COLL:GUID:CONN:PORT1 ""APC 3.5 female""", True
Ena.WriteString "SENS2:CORR:COLL:GUID:CONN:PORT2 ""APC 3.5 female""", True
Ena.WriteString "SENS2:CORR:COLL:GUID:CONN:PORT3 ""Not used"""
Ena.WriteString "SENS2:CORR:COLL:GUID:CONN:PORT4 ""Not used"""

Ena.WriteString "SENS2:CORR:COLL:GUID:CKIT:PORT1 'N4431B ECal 03605'", True
Ena.WriteString "SENS2:CORR:COLL:GUID:CKIT:PORT2 'N4431B ECal 03605'", True

    '    Ena.WriteString "SENS2:CORR:COLL:GUID:CKIT:PORT1 '" & Calkit(1) & "'", True
    '    Ena.WriteString "SENS2:CORR:COLL:GUID:CKIT:PORT2 '" & Calkit(1) & "'", True

Ena.WriteString "SENS2:CORR:COLL:GUID:INIT", True
Ena.WriteString "SENS2:CORR:COLL:GUID:STEPS?", True
numSteps = Ena.ReadNumber

For i = 1 To numSteps
   Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN" & CStr(i), True
Next

Ena.WriteString "SENS2:CORR:COLL:GUID:SAVE", True
```

Trigger

Unlike the E5071C, the E5080B does not have a Bus trigger. In the following example, the E5080B's trigger source is set to Manual and the trigger is executed with the INIT<ch>:IMM command.

E5071C example

```
'Trigger setup
Ena.WriteString ":TRIG:SOUR BUS", True
Ena.WriteString ":TRIG:SEQ:SCOP ALL", True     'Trig scope: All ch

' Ena.WriteString ":TRIG:SEQ:SCOP ACT", True   'Trig scope: Active ch

Ena.WriteString ":INIT1:CONT ON", True
Ena.WriteString ":INIT2:CONT ON", True
```

E5080B example

```
'Trigger setup
Ena.WriteString ":SENS1:SWEEP:MODE HOLD", True
Ena.WriteString ":SENS2:SWEEP:MODE HOLD", True
Ena.WriteString ":TRIG:SOUR MAN", True
```

E5071C example

```
'Making triggers
Ena.WriteString ":TRIG:SING", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

  'Use the following commands if Trig scope is Active ch.
  '   Ena.WriteString ":DISP:WIND1:ACT", True
  '   Ena.WriteString ":TRIG:SING", True
  '   Ena.WriteString "*OPC?", True
  '   Dummy = Ena.ReadNumber
  '   Ena.WriteString ":DISP:WIND2:ACT", True
  '   Ena.WriteString ":TRIG:SING", True
  '   Ena.WriteString "*OPC?", True
  '   Dummy = Ena.ReadNumber
```

E5080B example

```
'Making triggers
Ena.WriteString ":INIT1:IMM", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString ":INIT2:IMM", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

Querying measurement data

The following examples read the formatted data of the measurement traces to the arrays in the 64-bit binary format.

NOTE:

The size of formatted array data is different as follows;
E5071C Size=NOP x 2 in any display format
E5080B Size=NOP in the formats other than Smith/Polar, Size=NOP x 2 in Smith/Polar formats

E5071C example

```
'Preparations for trace data query
Dim Poin_ch1, Poin_ch2 As Integer
Dim Fdata_tr1_ch1() As Double, Fdata_tr2_ch1() As Double, Fdata_tr3_ch1() As Double
Dim Fdata_tr1_ch2() As Double, Fdata_tr2_ch2() As Double, Fdata_tr3_ch2() As Double

Ena.WriteString ":SENS1:SWE:POIN?", True
Poin_ch1 = Ena.ReadNumber
ReDim Fdata_tr1_ch1(Poin_ch1 * 2 - 1), Fdata_tr3_ch1(Poin_ch1 * 2 - 1), Fdata_tr3_ch1(Poin_ch1 * 2 - 1)

Ena.WriteString ":SENS2:SWE:POIN?", True
Poin_ch2 = Ena.ReadNumber
ReDim Fdata_tr1_ch2(Poin_ch1 * 2 - 1), Fdata_tr3_ch2(Poin_ch1 * 2 - 1), Fdata_tr3_ch2(Poin_ch1 * 2 - 1)

Ena.WriteString ":FORM:DATA REAL", True                    ' E5071C's "REAL" is 64-bit.
```

```
'Query formatted trace data

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:DATA:FDAT?", True
Fdata_tr1_ch1 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:DATA:FDAT?", True
Fdata_tr2_ch1 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:DATA:FDAT?", True
Fdata_tr3_ch1 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:DATA:FDAT?", True
Fdata_tr1_ch2 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:DATA:FDAT?", True
Fdata_tr2_ch2 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:DATA:FDAT?", True
Fdata_tr3_ch2 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

  'The following command is also available.
  '   Ena.WriteString ":CALC2:TRAC3:DATA:FDAT?", True
  '   Fdata_tr3_ch2 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)
```

E5080B example

```
'Preparations for trace data query
Dim Poin_ch1, Point_ch2 As Integer
Dim Fdat_tr1() As Double, Fdat_tr2() As Double, Fdat_tr3() As Double
Dim Fdat_tr4() As Double, Fdat_tr5() As Double, Fdat_tr6() As Double

Ena.WriteString ":SENS1:SWE:POIN?", True
Poin_ch1 = Ena.ReadNumber
ReDim Fdat_tr1(Poin_ch1 - 1), Fdat_tr3(Poin_ch1 - 1), Fdat_tr3(Poin_ch1 - 1)

Ena.WriteString ":SENS2:SWE:POIN?", True
Poin_ch2 = Ena.ReadNumber
ReDim Fdat_tr4(Poin_ch2 - 1), Fdat_tr5(Poin_ch2 - 1), Fdat_tr6(Poin_ch2 - 1)

Ena.WriteString ":FORM:DATA REAL,64", True                    ' "REAL,64" corresponds to E5071C's "REAL".
```

```
'Query formatted trace data

Ena.WriteString ":CALC1:MEAS1:DATA:FDAT?", True
Fdata_tr1 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC1:MEAS2:DATA:FDAT?", True
Fdata_tr2 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC1:MEAS3:DATA:FDAT?", True
Fdata_tr3 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:MEAS4:DATA:FDAT?", True
Fdata_tr4 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:MEAS5:DATA:FDAT?", True
Fdata_tr5 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

Ena.WriteString ":CALC2:MEAS6:DATA:FDAT?", True
Fdata_tr6 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)

  'Or the following commands can be used if 'Meas_x' has been defined before.
  '    Ena.WriteString ":CALC2:PAR:SEL 'Meas_6'", True
  '    Ena.WriteString ":CALC2:DATA? FDATA", True
  '    Fdata_tr6 = Ena.ReadIEEEBlock(BinaryType_R8, False, True)
```

NOTE:

E5080B's other trace data query commands

:CALC<cnum>:MEAS<mnum>:DATA:SDAT?	Queries the corrected data array of the trace specified by MEAS<mnum>.
:CALC<cnum>:DATA:MFData? <string>	Queries the formatted data of multiple traces specified by <string>.
:CALC<cnum>:DATA:MSData? <string>	Queries the corrected data array of multiple traces specified by <string>.

Saving / recalling instrument state & user cal data

E5071C example, save

```
Ena.WriteString ":MMEM:STOR:STYP CST", True
Ena.WriteString ":MMEM:STOR 'D:/StateCal_1.sta'", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

E5080B example, save

```
Ena.WriteString ":MMEM:STOR:CST 'D:/StateCal_1.cst'", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

E5071C example, recall

```
Ena.WriteString ":MMEM:LOAD 'D:/StateCal_1.sta'", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

E5080B example, recall

```
Ena.WriteString ":MMEM:LOAD 'D:/StateCal_1.cst'", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
```

Part 2: E5071C Code Emulation Mode

This chapter describes how to use the E5071C Code Emulation Mode that is available in the E5080B firmware rev.A.11.70 or above. In the E5071C Code Emulation Mode, the E5080B is remote controlled with the E5071C's SCPI commands instead of the E5080B's SCPI commands. If the E5071C's commands/functions you use are supported by this emulation mode, you can control the E5080B with your program with no or minor modifications. This enables you to migrate your E5071C to the E5080B more easily than rewriting the commands.

E5071C's functions supported by Code Emulation Mode

The following table summarizes the E5071C's functions which are supported by the E5071C Code Emulation Mode. For more details on the command-level support information, refer to the E5080B Help. (Visit www.keysight.com/find/E5080B to download it.)

Supported by E5071C Code Emulation Mode

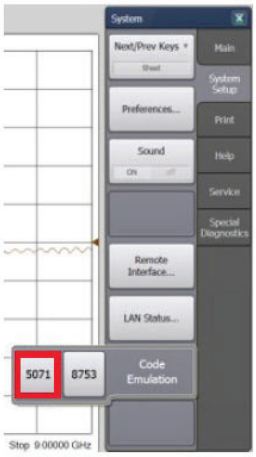
Trace/window/channel layout	All traces plotted in one window per channel, all channels displayed in one page All traces plotted in one window per channel, all channels split into sheets All traces split into different windows, all channels split into sheets
Sweep type	Linear, Log, Power, Segment
Calibration	**Automatically emulated** Mechanical full 1/2-port cal with defined thru, Mechanical response cal, Full 1/2/3/4-port ECal, Source power cal, Receiver power cal
	Executable by modifying program [1] Mechanical full 3/4-port cal, Mechanical unknown-thru cal, Enhanced response cal, Mixed-connector cal, TRL cal
Limit test	Limit line [2], Ripple limit, Bandwidth limit, Point limit [3]
Marker	Yes
Marker search	Max, Min, Peak, Target, Multi peak, Multi target, BW, Notch [4]
Equation editor	Yes [5]
Time domain / Gating	Yes
Frequency Offset Mode	No
External SG control	No
E5092A test set control	Yes
Save / Recall	Yes [6]
Status byte	No
VBA COM command	No

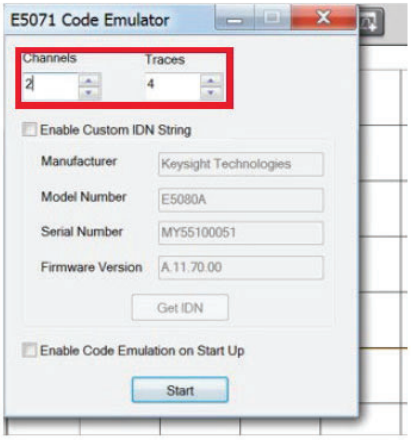
NOTE:
1. These calibrations can be executed by adding the E5080B's guided cal routine to the program.
2. Limit offset function is not available.
3. Not possible to perform both Limit line test and Point limit test for the same trace.
4. Not possible to perform both Bandwidth and Notch searches for the same trace.
5. Equations using parameters "data", "mem", and "xAxis" do not work.
6. In addition, the E5071C's state file can be converted to the E5080B's by using the E5071C State File Converter. (Visit www.keysight.com/find/E5080B to download it.)

Using Code Emulation Mode

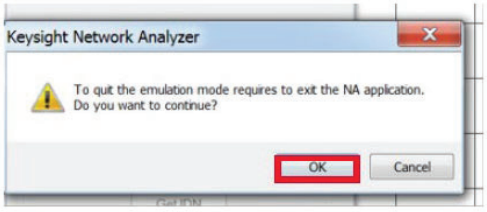
1. On the E5080B. press [System], “System Setup”, {Code Emulation}, and {5071}.
2. In the dialog box, set the emulator’s channel/trace to account for the number of channels and traces you are going to use in your program.

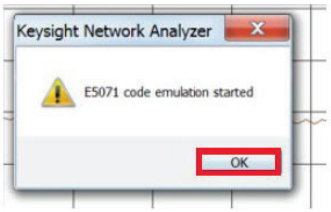
NOTE:
It is recommended to enter the number of channels that matches the number of channels you use in your program. In the following example, the number of channels is set to 2, matching the 2-channel filter measurement example discussed in this document. The number of traces is set to 4, and the trace number of the channel-2 starts from Tr5.



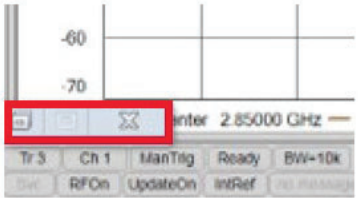


3. Press {OK} in the following dialogs.





Now the E5071C Code Emulation Mode has been launched, and you can remote control the E5080B with the E5071C’s SCPI commands. When the E5071C Code Emulation Mode is being executed, the emulator’s dialog appears in the lower left-hand corner of the screen. To finish the E5071C Code Emulation Mode and go back to the normal E5080B mode, close this dialog and restart the E5080B firmware by clicking its icon on the desktop.



The E5071C example program shown in pages 4 to 21 can be executed with no modification by using the E5071C Code Emulation Mode. However, there are some limitations in the trace/window/channel layout and the calibration, as described in the following sections.

Limitations of trace/window/channel layout

The trace/window/channel layouts available in the E5071C Code Emulation Mode are the following:

1. All traces plotted in one window per channel, and all channels displayed in one page
2. All traces plotted in one window per channel, and all channels split into sheets
3. All traces split into different windows, and all channels split into sheets

In the E5071C Code Emulation Mode, the trace/window/channel layout is determined by the following two E5071C commands. The final layout depends on which of these commands is executed later in the program.

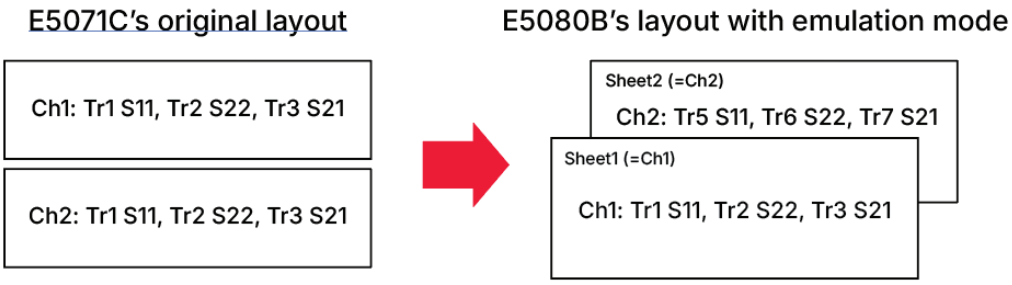
:DISP:WIND:SPL

This is the E5071C's command for allocating displayed traces. If the parameter specified by this command is other than "D1", all traces in the same channel are split into different windows. If the parameter is "D1", all traces in the same channel are plotted in the same window. The channels are split into different sheets in both cases.

:DISP:SPL

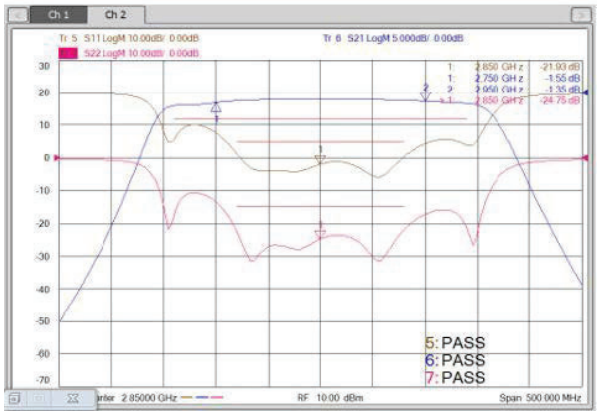
This is the E5071C's command for allocating displayed channels. If the parameter specified by this command is other than "D1", all traces in the same channel are plotted in the same window, and all channels are displayed in one page. If the parameter is "D1", all channels are split into different sheets.

This is the E5071C example routine that was shown on page 5. In this case, the code emulator plots all the traces of each channel in the same window but the channels are split into two sheets.

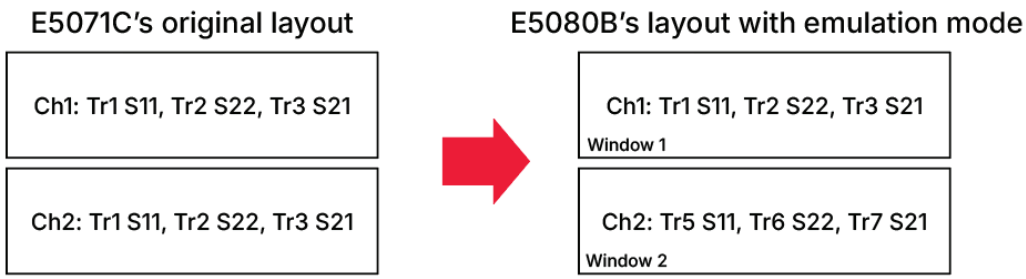


E5071C example

```
Ena.WriteString ":SYST:PRES", True

Ena.WriteString ":DISP:SPL D1_2", True

'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:PAR1:DEF S11", True
Ena.WriteString ":CALC1:PAR2:DEF S21", True
Ena.WriteString ":CALC1:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND1:SPL D1", True

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True

'Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:PAR1:DEF S11", True
Ena.WriteString ":CALC2:PAR2:DEF S21", True
Ena.WriteString ":CALC2:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND2:SPL D1", True

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
```



If you want to let the E5080B display all the channels in the same page, modify the source code to move the location of the command ":DISP:SPL D1_2" to the end, or delete the command ":DISP:WINDx:SPL D1".



E5071C example

Ena.WriteString ":SYST:PRES", True

~~Ena.WriteString ":DISP:SPL D1_2", True~~

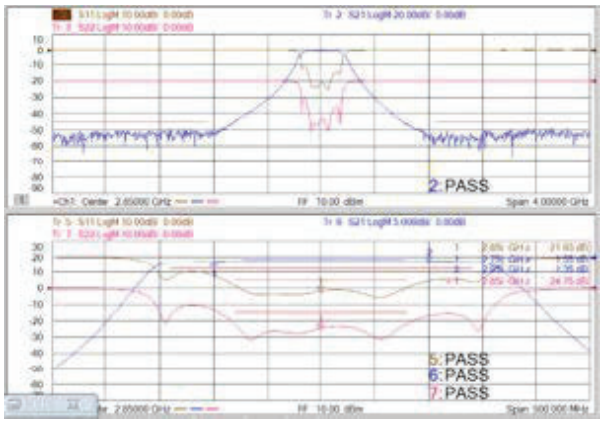
'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:PAR1:DEF S11", True
Ena.WriteString ":CALC1:PAR2:DEF S21", True
Ena.WriteString ":CALC1:PAR3:DEF S22", True
~~Ena.WriteString ":DISP:WIND1:SPL D1", True~~

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True

'Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:PAR1:DEF S11", True
Ena.WriteString ":CALC2:PAR2:DEF S21", True
Ena.WriteString ":CALC2:PAR3:DEF S22", True
~~Ena.WriteString ":DISP:WIND2:SPL D1", True~~

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True

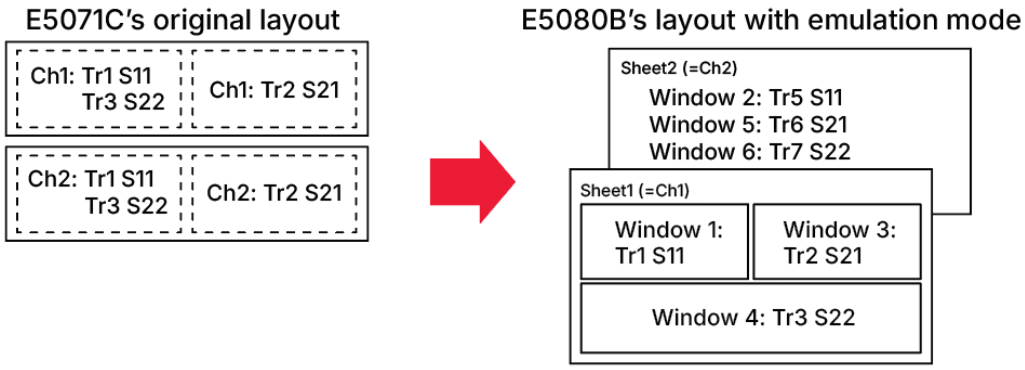
Ena.WriteString ":DISP:SPL D1_2", True



Or delete ":DISP:WINDx:SPL D1".

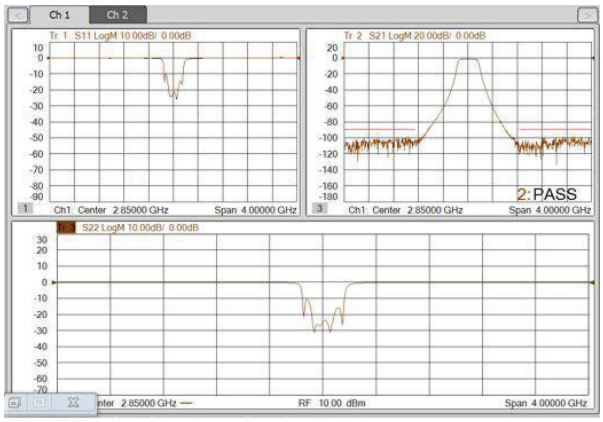
Move the location of ":DISP:SPL D1_2" to the end.

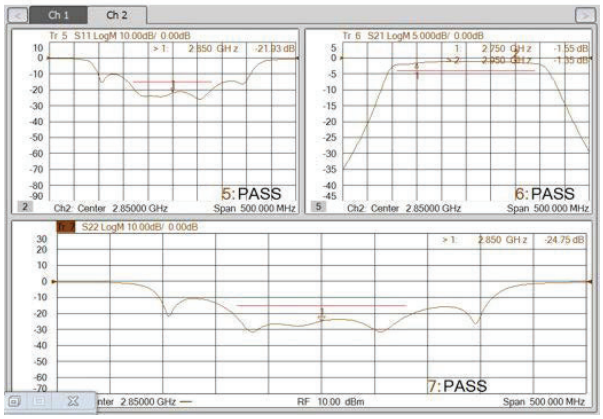
This is the E5071C example routine that was shown on the page 5. In this case, the code emulator splits all the traces of each channel into different windows and the channels are split into different sheets. If you change the parameter of the command ":DISP:WIND:SPL" from "D12" to "D1", all the traces will be plotted in the same window in each sheet.



E5071C example

```
Ena.WriteString ":SYST:PRES", True

Ena.WriteString ":DISP:SPL D1_2", True

'Ch-1 S-parameter and display format setup
Ena.WriteString ":CALC1:PAR:COUN 3", True
Ena.WriteString ":CALC1:PAR1:DEF S11", True
Ena.WriteString ":CALC1:PAR2:DEF S21", True
Ena.WriteString ":CALC1:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND1:SPL D12", True

Ena.WriteString ":CALC1:PAR1:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR2:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True
Ena.WriteString ":CALC1:PAR3:SEL", True
Ena.WriteString ":CALC1:SEL:FORM MLOG", True

'Ch-2 S-parameter and display format setup
Ena.WriteString ":CALC2:PAR:COUN 3", True
Ena.WriteString ":CALC2:PAR1:DEF S11", True
Ena.WriteString ":CALC2:PAR2:DEF S21", True
Ena.WriteString ":CALC2:PAR3:DEF S22", True
Ena.WriteString ":DISP:WIND2:SPL D12", True

Ena.WriteString ":CALC2:PAR1:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR2:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
Ena.WriteString ":CALC2:PAR3:SEL", True
Ena.WriteString ":CALC2:SEL:FORM MLOG", True
```





Limitation of calibration

This is an E5071C example routine for performing the mechanical full 2-port cal, which was shown on page 13. In this case, the code emulator can perform the full 2-port cal with no program modification. But the calibration is performed with the defined thru, because the code emulator internally uses the PNA's unguided cal commands that cannot perform the unknown thru calibration.

E5071C example

```
Dim Dummy As Long

Ena.WriteString ":SENS1:CORR:COLL:METH:SOLT2 1,2", True
Ena.WriteString ":SENS2:CORR:COLL:METH:SOLT2 1,2", True

Ena.WriteString ":SENS1:CORR:COLL:CKIT 1", True    '85033E
Ena.WriteString ":SENS2:CORR:COLL:CKIT 1", True

MsgBox ("Connect OPEN to port-1")
Ena.WriteString ":SENS1:CORR:COLL:OPEN 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:OPEN 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-1")
Ena.WriteString ":SENS1:CORR:COLL:SHOR 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:SHOR 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-1")
Ena.WriteString ":SENS1:CORR:COLL:LOAD 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:LOAD 1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect OPEN to port-2")
Ena.WriteString ":SENS1:CORR:COLL:OPEN 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:OPEN 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-2")
Ena.WriteString ":SENS1:CORR:COLL:SHOR 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:SHOR 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-2")
Ena.WriteString ":SENS1:CORR:COLL:LOAD 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:LOAD 2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect THRU between port-1 & 2")
Ena.WriteString ":SENS1:CORR:COLL:THRU 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS1:CORR:COLL:THRU 2,1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:THRU 1,2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString ":SENS2:CORR:COLL:THRU 2,1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString ":SENS1:CORR:COLL:SAVE", True
Ena.WriteString ":SENS2:CORR:COLL:SAVE", True
```

To perform the unknown thru full 2-port cal, or other advanced calibrations such as the full 3/4port cal, modify your program to remove the above original calibration routine and add the new calibration routine that uses the E5080B's guided calibration commands. The guided cal commands are the only E5080B commands that can be mixed with the E5071C's commands in the code emulation mode. See the following examples for how to perform the unknown thru full 2-port cal.

This is an example of a routine that can be added to the original E5071C program to perform the mechanical unknown thru full 2-port cal (also shown on page 15).

Unknown-thru full 2-port cal using E5080B's commands (A)

```
Dim Dummy As Long

Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:INIT", True

Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:INIT", True

MsgBox ("Connect OPEN to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN1", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN2", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-1")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN3", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN3", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect OPEN to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN4", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN4", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect SHORT to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN5", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN5", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect LOAD to port-2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN6", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN6", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

MsgBox ("Connect THRU betwen port-1 & 2")
Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN7", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber
Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN7", True
Ena.WriteString "*OPC?", True
Dummy = Ena.ReadNumber

Ena.WriteString "SENS1:CORR:COLL:GUID:SAVE", True
Ena.WriteString "SENS2:CORR:COLL:GUID:SAVE", True
```

This is another example of a routine that can be added to the original E5071C program to perform the mechanical unknown thru full 2-port cal (also shown on page 17).

Unknown-thru full 2-port cal using E5080B's commands (B)

```
Dim numSteps As Integer, strPrompt As String
Dim i As Integer

   'You can confirm the names of supported cal kits as follows.
   '    Dim Connector, Calkit As String
   '    Ena.WriteString "SENS1:CORR:COLL:GUID:CONN:CAT?"
   '    Connector = Ena.ReadString
   '    MsgBox Connector
   '    Ena.WriteString "SENS1:CORR:COLL:GUID:CKIT:CAT? 'APC 3.5 female'"
   '    Calkit = Ena.ReadString
   '    MsgBox Calkit

Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS1:CORR:COLL:GUID:INIT", True

Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT1 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CONN:PORT2 'APC 3.5 female'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT1 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:CKIT:PORT2 '85033D/E'", True
Ena.WriteString ":SENS2:CORR:COLL:GUID:INIT", True

Ena.WriteString ":SENS1:CORR:COLL:GUID:STEPS?", True
numSteps = Ena.ReadNumber

For i = 1 To numSteps
   Ena.WriteString "SENS1:CORR:COLL:GUID:DESC? " + CStr(i), True
   strPrompt = Ena.ReadString
   MsgBox strPrompt
   Ena.WriteString "SENS1:CORR:COLL:GUID:ACQ STAN" + CStr(i), True
   Ena.WriteString "SENS2:CORR:COLL:GUID:ACQ STAN" + CStr(i), True
Next

Ena.WriteString "SENS1:CORR:COLL:GUID:SAVE", True
Ena.WriteString "SENS2:CORR:COLL:GUID:SAVE", True
```

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

KEYSIGHT

This information is subject to change without notice. © Keysight Technologies, 2019 – 2026, Published in USA, March 10, 2026, 5992-3873EN