

**Agilent 35670A Dynamic Signal Analyzer**

**GPIB Programming  
with the Agilent 35670A**

Agilent Part Number 35670-90057  
Print Date: December 2000

Copyright © Agilent Technologies, Inc., 1991, 1992, 1994, 1995, 2000. All rights reserved.  
8600 Soper Hill Road Everett, Washington 98205-1209 U.S.A.

---

## ***Safety Summary***

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Agilent Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

### **GENERAL**

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1.

### **ENVIRONMENTAL CONDITIONS**

This instrument is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters. Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

### **BEFORE APPLYING POWER**

Verify that the product is set to match the available line voltage, the correct fuse is installed, and all safety precautions are taken. Note the instrument's external markings described under Safety Symbols.

### **GROUND THE INSTRUMENT**

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

## **FUSES**

Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuse holders. To do so could cause a shock or fire hazard.

## **DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE**

Do not operate the instrument in the presence of flammable gases or fumes.

## **DO NOT REMOVE THE INSTRUMENT COVER**

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified service personnel.

Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

---

### **WARNING**

---

**The WARNING sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.**

---

### **Caution**

---

The CAUTION sign denotes a hazard. It calls attention to an operating procedure, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a CAUTION sign until the indicated conditions are fully understood and met.

## Safety Symbols



Warning, risk of electric shock

---



Caution, refer to accompanying documents

---



Alternating current

---



Both direct and alternating current

---



Earth (ground) terminal

---



Protective earth (ground) terminal

---



Frame or chassis terminal

---



Terminal is at earth potential.

---



Standby (supply). Units with this symbol are not completely disconnected from ac mains when this switch is off

# TABLE OF CONTENTS

---

## CHAPTER 1: GPIB Programming with the Agilent 35670A

GPIB Setup . . . . .	1-2
Configuring the GPIB System . . . . .	1-2
Quick Verification . . . . .	1-5
Verification Program . . . . .	1-7
How the Agilent 35670A Operates in an GPIB System . . . . .	1-8
Controller Capabilities . . . . .	1-8
GPIB Commands That Require Passing Control to the Agilent 35670A . . . . .	1-8
GPIB Interface Capabilities. . . . .	1-9
GPIB Queues in the Agilent 35670A. . . . .	1-10
Command Synchronization . . . . .	1-10
Overlapped Commands in the Agilent 35670A . . . . .	1-11
GPIB Commands That Transfer Mixed (ASCII/REAL) Data . . . . .	1-12
Determining Units . . . . .	1-12
The Agilent 35670A's Status Registers . . . . .	1-13
Register Summary . . . . .	1-13
Status Byte Register Set . . . . .	1-14
Device State Register Set . . . . .	1-16
Limit Fail Register Set . . . . .	1-17
Questionable Status Register Set . . . . .	1-18
Questionable Voltage Register Set . . . . .	1-19
Standard Event Register Set . . . . .	1-20
Operation Status Register Set. . . . .	1-22
User Status Register Set . . . . .	1-24
Agilent 35670A Register Set Summary . . . . .	1-25
SCPI Compliance Information . . . . .	1-26
Confirmed Commands . . . . .	1-27
List of Approved Commands . . . . .	1-31
List of Instrument Specific Commands. . . . .	1-32

## CHAPTER 2: Introduction to the Command Reference

Finding the Right Command . . . . .	2-3
Command Syntax in the Agilent 35670A . . . . .	2-5
Special Syntactic Elements . . . . .	2-5
Conventions. . . . .	2-6
Syntax Descriptions. . . . .	2-7

CHAPTER 3: Common Commands	
*CAL?query . . . . .	3-2
*CLScommand . . . . .	3-3
*ESEcommand/query . . . . .	3-4
*ESR?query . . . . .	3-5
*IDN?query . . . . .	3-6
*OPCcommand/query . . . . .	3-7
*OPT?query . . . . .	3-9
*PCBcommand . . . . .	3-10
*PSCcommand/query . . . . .	3-11
*RSTcommand . . . . .	3-12
*SREcommand/query . . . . .	3-14
*STB?query . . . . .	3-16
*TRGcommand . . . . .	3-18
*TST?query . . . . .	3-19
*WAIcommand . . . . .	3-20
CHAPTER 4: ABORt	
ABORtcommand . . . . .	4-2
CHAPTER 5: ARM	
ARM[:IMMediate]command . . . . .	5-2
ARM:RPM:INCRementcommand/query . . . . .	5-3
ARM:RPM:MODEcommand/query . . . . .	5-4
ARM:RPM:THResholdcommand/query . . . . .	5-5
ARM:SOURcecommand/query . . . . .	5-6
ARM:TIMercommand/query . . . . .	5-8
CHAPTER 6: CALCulate	
CALCulate[1 2 3 4]:ACTivecommand/query . . . . .	6-3
CALCulate[1 2 3 4]:CFIT:ABORtcommand . . . . .	6-4
CALCulate[1 2 3 4]:CFIT:COPIcommand . . . . .	6-5
CALCulate[1 2 3 4]:CFIT:DATAcommand/query . . . . .	6-6
CALCulate[1 2 3 4]:CFIT:DESTinationcommand/query . . . . .	6-8
CALCulate[1 2 3 4]:CFIT:FREQUency[:AUTO]command/query . . . . .	6-9
CALCulate[1 2 3 4]:CFIT:FREQUency:STARtcommand/query . . . . .	6-10
CALCulate[1 2 3 4]:CFIT:FREQUency:STOPcommand/query . . . . .	6-11
CALCulate[1 2 3 4]:CFIT:FSCalecommand/query . . . . .	6-12
CALCulate[1 2 3 4]:CFIT[:IMMediate]command . . . . .	6-13
CALCulate[1 2 3 4]:CFIT:ORDer:AUTOcommand/query . . . . .	6-14
CALCulate[1 2 3 4]:CFIT:ORDer:POLescommand/query . . . . .	6-15
CALCulate[1 2 3 4]:CFIT:ORDer:ZERoscommand/query . . . . .	6-16
CALCulate[1 2 3 4]:CFIT:TDELaycommand/query . . . . .	6-17
CALCulate[1 2 3 4]:CFIT:WEIGHt:AUTOcommand/query . . . . .	6-18
CALCulate[1 2 3 4]:CFIT:WEIGHt:REGistercommand/query . . . . .	6-19
CALCulate[1 2 3 4]:DATA?query . . . . .	6-20
CALCulate[1 2 3 4]:DATA:HEADer:POINts query . . . . .	6-22
CALCulate[1 2 3 4]:FEEDcommand/query . . . . .	6-23
CALCulate[1 2 3 4]:FORMatcommand/query . . . . .	6-29
CALCulate[1 2 3 4]:GDAPerture:APERturecommand/query . . . . .	6-31
CALCulate[1 2 3 4]:LIMit:BEEP[:STATe]command/query . . . . .	6-33
CALCulate[1 2 3 4]:LIMit:FAIL?query . . . . .	6-34
CALCulate[1 2 3 4]:LIMit:LOWer:CLEar[:IMMediate]command . . . . .	6-35
CALCulate[1 2 3 4]:LIMit:LOWer:MOVE:Ycommand . . . . .	6-36
CALCulate[1 2 3 4]:LIMit:LOWer:REPor[:DATA]?query . . . . .	6-37
CALCulate[1 2 3 4]:LIMit:LOWer:REPor:YDATA?query . . . . .	6-38
CALCulate[1 2 3 4]:LIMit:LOWer:SEGMENTcommand/query . . . . .	6-39

CALCulate[1 2 3 4]:LIMit:LOWer:SEGment:CLEarcommand . . . . .	6-41
CALCulate[1 2 3 4]:LIMit:LOWer:TRACe[:IMMediate]command . . . . .	6-42
CALCulate[1 2 3 4]:LIMit:STATecommand/query . . . . .	6-43
CALCulate[1 2 3 4]:LIMit:UPPer:CLEar[:IMMediate]command . . . . .	6-44
CALCulate[1 2 3 4]:LIMit:UPPer:MOVE:Ycommand . . . . .	6-45
CALCulate[1 2 3 4]:LIMit:UPPer:REPort[:DATA]?query . . . . .	6-46
CALCulate[1 2 3 4]:LIMit:UPPer:REPort:YDATA?query . . . . .	6-47
CALCulate[1 2 3 4]:LIMit:UPPer:SEGmentcommand/query . . . . .	6-48
CALCulate[1 2 3 4]:LIMit:UPPer:SEGment:CLEarcommand . . . . .	6-50
CALCulate[1 2 3 4]:LIMit:UPPer:TRACe[:IMMediate]command . . . . .	6-51
CALCulate[1 2 3 4]:MARKer:BAND:STARtcommand/query . . . . .	6-52
CALCulate[1 2 3 4]:MARKer:BAND:STOPcommand/query . . . . .	6-54
CALCulate[1 2 3 4]:MARKer:COUPled[:STATe]command/query . . . . .	6-56
CALCulate[1 2 3 4]:MARKer:DTABLE:CLEar[:IMMediate]command . . . . .	6-57
CALCulate[1 2 3 4]:MARKer:DTABLE:COPI[1 2 3 4]command . . . . .	6-58
CALCulate[1 2 3 4]:MARKer:DTABLE[:DATA]?query . . . . .	6-59
CALCulate[1 2 3 4]:MARKer:DTABLE:X[:DATA]?query . . . . .	6-61
CALCulate[1 2 3 4]:MARKer:DTABLE:X:DELetecommand . . . . .	6-62
CALCulate[1 2 3 4]:MARKer:DTABLE:X:INSertcommand/query . . . . .	6-63
CALCulate[1 2 3 4]:MARKer:DTABLE:X:LABelcommand/query . . . . .	6-64
CALCulate[1 2 3 4]:MARKer:DTABLE:X:SELect[:POINT]command/query . . . . .	6-65
CALCulate[1 2 3 4]:MARKer:FUNCTioncommand/query . . . . .	6-66
CALCulate[1 2 3 4]:MARKer:FUNCTion:RESult?query . . . . .	6-71
CALCulate[1 2 3 4]:MARKer:HARMonic:COUNTcommand/query . . . . .	6-73
CALCulate[1 2 3 4]:MARKer:HARMonic:FUNDamentalcommand/query . . . . .	6-74
CALCulate[1 2 3 4]:MARKer:MAXimum[:GLOBal]command . . . . .	6-75
CALCulate[1 2 3 4]:MARKer:MAXimum[:GLOBal]:TRACkcommand/query . . . . .	6-76
CALCulate[1 2 3 4]:MARKer:MAXimum:LEFTcommand . . . . .	6-77
CALCulate[1 2 3 4]:MARKer:MAXimum:RIGHTcommand . . . . .	6-78
CALCulate[1 2 3 4]:MARKer:MODEcommand/query . . . . .	6-79
CALCulate[1 2 3 4]:MARKer:POSitioncommand/query . . . . .	6-80
CALCulate[1 2 3 4]:MARKer:POSition:POINTcommand/query . . . . .	6-81
CALCulate[1 2 3 4]:MARKer:REFerence:Xcommand/query . . . . .	6-84
CALCulate[1 2 3 4]:MARKer:REFerence:Ycommand/query . . . . .	6-85
CALCulate[1 2 3 4]:MARKer:SIDeband:CARRiercommand/query . . . . .	6-86
CALCulate[1 2 3 4]:MARKer:SIDeband:COUNTcommand/query . . . . .	6-87
CALCulate[1 2 3 4]:MARKer:SIDeband:INCRementcommand/query . . . . .	6-88
CALCulate[1 2 3 4]:MARKer[:STATe]command/query . . . . .	6-89
CALCulate[1 2 3 4]:MARKer:X[:ABSolute]command/query . . . . .	6-90
CALCulate[1 2 3 4]:MARKer:X:RELativecommand/query . . . . .	6-91
CALCulate[1 2 3 4]:MARKer:Y[:ABSolute]?query . . . . .	6-92
CALCulate[1 2 3 4]:MARKer:Y:RELativecommand/query . . . . .	6-93
CALCulate[1 2 3 4]:MATH:CONStant[1 2 3 4 5]command/query . . . . .	6-94
CALCulate[1 2 3 4]:MATH:DATAcommand/query . . . . .	6-95
CALCulate[1 2 3 4]:MATH[:EXPRession[1 2 3 4 5]]command/query . . . . .	6-97
CALCulate[1 2 3 4]:MATH:SELectcommand/query . . . . .	6-99
CALCulate[1 2 3 4]:MATH:STATecommand/query . . . . .	6-100
CALCulate[1 2 3 4]:SYNTHeSis:COPIcommand . . . . .	6-101
CALCulate[1 2 3 4]:SYNTHeSis:DATAcommand/query . . . . .	6-102
CALCulate[1 2 3 4]:SYNTHeSis:DESTinationcommand/query . . . . .	6-104
CALCulate[1 2 3 4]:SYNTHeSis:FSCalecommand/query . . . . .	6-105
CALCulate[1 2 3 4]:SYNTHeSis:GAINcommand/query . . . . .	6-106
CALCulate[1 2 3 4]:SYNTHeSis[:IMMediate]command . . . . .	6-107
CALCulate[1 2 3 4]:SYNTHeSis:SPACingcommand/query . . . . .	6-108

CALCulate[1 2 3 4]:SYNThesis:TDELAycommand/query . . . . .	6-109
CALCulate[1 2 3 4]:SYNThesis:TTYPecommand/query . . . . .	6-110
CALCulate[1 2 3 4]:UNIT:AMPLitudecommand/query . . . . .	6-111
CALCulate[1 2 3 4]:UNIT:ANGLecommand/query . . . . .	6-113
CALCulate[1 2 3 4]:UNIT:DBReferencecommand/query . . . . .	6-114
CALCulate[1 2 3 4]:UNIT:DBReference:IMPedancecommand/query . . . . .	6-115
CALCulate[1 2 3 4]:UNIT:DBReference:USER:LABelcommand/query . . . . .	6-116
CALCulate[1 2 3 4]:UNIT:DBReference:USER:REFerencecommand/query . . . . .	6-117
CALCulate[1 2 3 4]:UNIT:MECHanicalcommand/query . . . . .	6-118
CALCulate[1 2 3 4]:UNIT:VOLTagecommand/query . . . . .	6-119
CALCulate[1 2 3 4]:UNIT:Xcommand/query . . . . .	6-121
CALCulate[1 2 3 4]:UNIT:X:ORDer:FACTorcommand/query . . . . .	6-123
CALCulate[1 2 3 4]:UNIT:X:USER:FREQuency:FACTorcommand/query . . . . .	6-124
CALCulate[1 2 3 4]:UNIT:X:USER:FREQuency:LABelcommand/query . . . . .	6-125
CALCulate[1 2 3 4]:UNIT:X:USER:TIME:FACTorcommand/query . . . . .	6-126
CALCulate[1 2 3 4]:UNIT:X:USER:TIME:LABelcommand/query . . . . .	6-127
CALCulate[1 2 3 4]:WATERfall:COUNTcommand/query . . . . .	6-128
CALCulate[1 2 3 4]:WATERfall[:DATA]?query . . . . .	6-129
CALCulate[1 2 3 4]:WATERfall:SLICE:COPIcommand . . . . .	6-131
CALCulate[1 2 3 4]:WATERfall:SLICE:SELEctcommand/query . . . . .	6-132
CALCulate[1 2 3 4]:WATERfall:SLICE:SELEct:POINtcommand/query . . . . .	6-133
CALCulate[1 2 3 4]:WATERfall:TRACe:COPIcommand . . . . .	6-135
CALCulate[1 2 3 4]:WATERfall:TRACe:SELEctcommand/query . . . . .	6-136
CALCulate[1 2 3 4]:WATERfall:TRACe:SELEct:POINtcommand/query . . . . .	6-137
CALCulate[1 2 3 4]:X:DATA?query . . . . .	6-138
CHAPTER 7: CALibration	
CALibration[:ALL]?query . . . . .	7-2
CALibration:AUTOcommand/query . . . . .	7-3
CHAPTER 8: DISPlay	
DISPlay:ANNotation[:ALL]command/query . . . . .	8-2
DISPlay:BODEcommand . . . . .	8-3
DISPlay:BRIGHtnesscommand/query . . . . .	8-4
DISPlay:ERRorcommand . . . . .	8-5
DISPlay:EXTernal[:STATe]command/query . . . . .	8-6
DISPlay:FORMatcommand/query . . . . .	8-7
DISPlay:GPIB:ECHOcommand/query . . . . .	8-8
DISPlay:PROGram:KEY:BOXcommand/query . . . . .	8-9
DISPlay:PROGram:KEY:BRACketcommand/query . . . . .	8-10
DISPlay:PROGram[:MODE]ommand/query . . . . .	8-11
DISPlay:PROGram:VECTor:BUFFer[:STATe]command/query . . . . .	8-12
DISPlay:RPM[:STATe]command/query . . . . .	8-13
DISPlay:SHOWall[:STATe]command/query . . . . .	8-14
DISPlay:STATecommand/query . . . . .	8-15
DISPlay:TCAPture:ENVELOpe[:STATe]command/query . . . . .	8-16
DISPlay:VIEWcommand/query . . . . .	8-17
DISPlay[:WINDow[1 2 3 4]]:DTABLE:MARKer[:STATe]command/query . . . . .	8-18
DISPlay[:WINDow[1 2 3 4]]:DTABLE[:STATe]command/query . . . . .	8-19
DISPlay[:WINDow[1 2 3 4]]:LIMit:STATecommand/query . . . . .	8-20
DISPlay[:WINDow[1 2 3 4]]:POLar:CLOCKwiseommand/query . . . . .	8-21
DISPlay[:WINDow[1 2 3 4]]:POLar:ROTationcommand/query . . . . .	8-22
DISPlay[:WINDow[1 2 3 4]]:TRACe:APOWer[:STATe]command/query . . . . .	8-24
DISPlay[:WINDow[1 2 3 4]]:TRACe:BPOWer[:STATe]command/query . . . . .	8-25
DISPlay[:WINDow[1 2 3 4]]:TRACe:GRATICule:GRID[:STATe]command/query . . . . .	8-26
DISPlay[:WINDow[1 2 3 4]]:TRACe:LABelcommand/query . . . . .	8-27



DISPlay[:WINDow[1 2 3 4]]:TRACe:LABel:DEFault[:STATe]command/query . . . . .	8-28
DISPlay[:WINDow[1 2 3 4]]:TRACe:X:MATCh[1 2 3 4]command . . . . .	8-29
DISPlay[:WINDow[1 2 3 4]]:TRACe:X[:SCALe]:AUTOcommand/query. . . . .	8-30
DISPlay[:WINDow[1 2 3 4]]:TRACe:X[:SCALe]:LEFTcommand/query . . . . .	8-31
DISPlay[:WINDow[1 2 3 4]]:TRACe:X[:SCALe]:RIGHtcommand/query. . . . .	8-32
DISPlay[:WINDow[1 2 3 4]]:TRACe:X:SPACingcommand/query . . . . .	8-33
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y:MATCh[1 2 3 4]command . . . . .	8-34
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:AUTOcommand/query. . . . .	8-35
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:BOTTOmcommand/query . . . . .	8-36
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:CENTercommand/query. . . . .	8-37
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:PDIVisioncommand/query . . . . .	8-38
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:REFerencecommand/query . . . . .	8-39
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y[:SCALe]:TOPcommand/query . . . . .	8-41
DISPlay[:WINDow[1 2 3 4]]:TRACe:Y:SPACingommand/query . . . . .	8-42
DISPlay[:WINDow[1 2 3 4]]:WATERfall:BASElinecommand/query . . . . .	8-43
DISPlay[:WINDow[1 2 3 4]]:WATERfall:BOTTOmcommand/query . . . . .	8-44
DISPlay[:WINDow[1 2 3 4]]:WATERfall:COUNTcommand/query . . . . .	8-45
DISPlay[:WINDow[1 2 3 4]]:WATERfall:HEIGHtcommand/query . . . . .	8-46
DISPlay[:WINDow[1 2 3 4]]:WATERfall:HIDDENcommand/query. . . . .	8-47
DISPlay[:WINDow[1 2 3 4]]:WATERfall:SKEWcommand/query . . . . .	8-48
DISPlay[:WINDow[1 2 3 4]]:WATERfall:SKEW:ANGLecommand/query. . . . .	8-49
DISPlay[:WINDow[1 2 3 4]]:WATERfall[:STATe]command/query . . . . .	8-50
DISPlay[:WINDow[1 2 3 4]]:WATERfall:TOPcommand/query. . . . .	8-51
CHAPTER 9: FORMat	
FORMat[:DATA]command/query . . . . .	9-2
CHAPTER 10: HCOpy	
HCOpy:COLor:DEFaultcommand . . . . .	10-2
HCOpy:DESTinationcommand/query . . . . .	10-3
HCOpy:DEvice:LANGUagecommand/query. . . . .	10-4
HCOpy:DEvice:RESolutioncommand/query. . . . .	10-5
HCOpy:DEvice:SPEedcommand/query . . . . .	10-6
HCOpy[:IMMediate]command . . . . .	10-7
HCOpy:ITEM:FFEed:STATecommand/query . . . . .	10-8
HCOpy:ITEM:LABel:COLorcommand/query . . . . .	10-9
HCOpy:ITEM:LABel:STATecommand/query . . . . .	10-10
HCOpy:ITEM:LABel:TEXTcommand/query . . . . .	10-11
HCOpy:ITEM:TDSTamp:FORMatcommand/query . . . . .	10-12
HCOpy:ITEM:TDSTamp:STATecommand/query . . . . .	10-13
HCOpy:ITEM[:WINDow[1 2 3 4]]:TRACe:COLorcommand/query. . . . .	10-14
HCOpy:ITEM[:WINDow[1 2 3 4]]:TRACe:GRATICule:COLorcommand/query . . . . .	10-15
HCOpy:ITEM[:WINDow[1 2 3 4]]:TRACe:LIMit:LYPEcommand/query . . . . .	10-16
HCOpy:ITEM[:WINDow[1 2 3 4]]:TRACe:LYPEcommand/query. . . . .	10-17
HCOpy:ITEM[:WINDow[1 2 3 4]]:TRACe:MARKer:COLorcommand/query . . . . .	10-18
HCOpy:PAGE:DIMensions:AUTOcommand/query . . . . .	10-19
HCOpy:PAGE:DIMensions:USER:LLEFtommand/query . . . . .	10-20
HCOpy:PAGE:DIMensions:USER:URIGHtcommand/query. . . . .	10-22
HCOpy:PLOT:ADDReSScommand/query . . . . .	10-23
HCOpy:PRINt:ADDReSScommand/query . . . . .	10-24
HCOpy:TITLe[1 2]command/query. . . . .	10-25
HCOpy:SOURcecommand/query . . . . .	10-26

CHAPTER 11: INITiate	
INITiate:CONTInuouscommand/query . . . . .	11-2
INITiate[:IMMediate]command. . . . .	11-3
CHAPTER 12: INPut	
INPut[1 2 3 4]:BIAS[:STATe]command/query . . . . .	12-2
INPut[1 2 3 4]:COUPlingcommand/query. . . . .	12-3
INPut[1 2 3 4]:FILTer:AWEighting[:STATe]command/query . . . . .	12-4
INPut[1 2 3 4]:FILTer[:LPASs][:STATe]command/query . . . . .	12-5
INPut[1 2 3 4]:LOWcommand/query . . . . .	12-6
INPut[1 2 3 4]:REFerence:DIRectioNcommand/query . . . . .	12-7
INPut[1 2 3 4]:REFerence:POINtcommand/query . . . . .	12-8
INPut[1 2 3 4][:STATe]command/query . . . . .	12-9
CHAPTER 13: INSTRument	
INSTRument:NSElectcommand/query . . . . .	13-2
INSTRument[:SElect]command/query . . . . .	13-3
CHAPTER 14: MEMory	
MEMory:CATalog[:ALL]?query . . . . .	14-2
MEMory:CATalog:NAME?query. . . . .	14-3
MEMory:DELeTe:ALLcommand . . . . .	14-4
MEMory:DELeTe[:NAME]command . . . . .	14-5
MEMory:FREE[:ALL]?query. . . . .	14-6
CHAPTER 15: MMEMory	
MMEMory:COpycommand . . . . .	15-3
MMEMory:DELeTeommand . . . . .	15-4
MMEMory:DISK:ADDReSScommand/query. . . . .	15-5
MMEMory:DISK:UNITcommand/query . . . . .	15-6
MMEMory:FSYSstem?query . . . . .	15-7
MMEMory:INITializecommand . . . . .	15-8
MMEMory:LOAD:CFITcommand . . . . .	15-10
MMEMory:LOAD:CONTInuecommand/query. . . . .	15-11
MMEMory:LOAD:DTABLE:TRACe[1 2 3 4]command . . . . .	15-12
MMEMory:LOAD:LIMit:LOWer:TRACe[1 2 3 4]command . . . . .	15-13
MMEMory:LOAD:LIMit:UPPer:TRACe[1 2 3 4]command . . . . .	15-14
MMEMory:LOAD:MATHcommand . . . . .	15-15
MMEMory:LOAD:PROGramcommand . . . . .	15-16
MMEMory:LOAD:STATecommand . . . . .	15-17
MMEMory:LOAD:SYNThesiscommand . . . . .	15-18
MMEMory:LOAD:TCAPturecommand . . . . .	15-19
MMEMory:LOAD:TRACecommand. . . . .	15-20
MMEMory:LOAD:WATERfallcommand . . . . .	15-21
MMEMory:MDIRectorycommand . . . . .	15-22
MMEMory:MOVEcommand . . . . .	15-23
MMEMory:MSIScommand/query . . . . .	15-24
MMEMory:NAMEcommand/query. . . . .	15-25
MMEMory:STORE:CFITcommand. . . . .	15-26
MMEMory:STORE:CONTInuecommand/query . . . . .	15-27
MMEMory:STORE:DTABLE:TRACe[1 2 3 4]command . . . . .	15-28
MMEMory:STORE:LIMit:LOWer:TRACe[1 2 3 4]command . . . . .	15-29
MMEMory:STORE:LIMit:UPPer:TRACe[1 2 3 4]command. . . . .	15-30
MMEMory:STORE:MATHcommand. . . . .	15-31
MMEMory:STORE:PROGramommand. . . . .	15-32
MMEMory:STORE:PROGram:FORMatcommand/query . . . . .	15-33
MMEMory:STORE:STATecommand. . . . .	15-34
MMEMory:STORE:SYNThesiscommand . . . . .	15-35

MMEMory:STORE:TCAPturecommand . . . . .	15-36
MMEMory:STORE:TRACecommand . . . . .	15-37
MMEM:STORE:TRACe:FORMatcommand/query . . . . .	15-38
MMEMory:STORE:WATERfallcommand . . . . .	15-39
CHAPTER 16: OUTPut	
OUTPut:FILTer[:LPASs][:STATe]command/query . . . . .	16-2
OUTPut[:STATe]command/query . . . . .	16-3
CHAPTER 17: PROGram	
PROGram:EDIT:ENABLecommand/query . . . . .	17-2
PROGram:EXPLicit:DEFinecommand/query . . . . .	17-3
PROGram:EXPLicit:LABelcommand/query . . . . .	17-4
PROGram[:SElected]:DEFinecommand/query . . . . .	17-5
PROGram[:SElected]:DELeTe:ALLcommand . . . . .	17-6
PROGram[:SElected]:DELeTe[:SElected]command . . . . .	17-7
PROGram[:SElected]:LABelcommand/query . . . . .	17-8
PROGram[:SElected]:MALLocatecommand/query . . . . .	17-9
PROGram[:SElected]:NAMEcommand/query . . . . .	17-10
PROGram[:SElected]:NUMBERcommand/query . . . . .	17-11
PROGram[:SElected]:STATecommand/query . . . . .	17-12
PROGram[:SElected]:STRingcommand/query . . . . .	17-13
CHAPTER 18: [SENSe:]	
[SENSe:]AVERAge:CONFidencecommand/query . . . . .	18-2
[SENSe:]AVERAge:COUNTcommand/query . . . . .	18-3
[SENSe:]AVERAge:HOLDcommand/query . . . . .	18-4
[SENSe:]AVERAge:IMPulsecommand/query . . . . .	18-5
[SENSe:]AVERAge:IREsult:RATEcommand/query . . . . .	18-6
[SENSe:]AVERAge:IREsult[:STATe]command/query . . . . .	18-7
[SENSe:]AVERAge:PREViewcommand/query . . . . .	18-8
[SENSe:]AVERAge:PREView:ACCeptcommand . . . . .	18-9
[SENSe:]AVERAge:PREView:REJectcommand . . . . .	18-10
[SENSe:]AVERAge:PREView:TIMEcommand/query . . . . .	18-11
[SENSe:]AVERAge[:STATe]and/query . . . . .	18-12
[SENSe:]AVERAge:TCONtrolcommand/query . . . . .	18-14
[SENSe:]AVERAge:TIMEcommand/query . . . . .	18-17
[SENSe:]AVERAge:TYPEcommand/query . . . . .	18-19
[SENSe:]DATAcommand/query . . . . .	18-21
[SENSe:]DATA:HEADer:FREQuency:START?query . . . . .	18-22
[SENSe:]DATA:HEADer:FREQuency:STOP?query . . . . .	18-23
[SENSe:]DATA:HEADer:POINtsquery . . . . .	18-24
[SENSe:]DATA:RANGecommand/query . . . . .	18-25
[SENSe:]FEEDcommand/query . . . . .	18-26
[SENSe:]FREQuency:BLocksizecommand/query . . . . .	18-27
[SENSe:]FREQuency:CENTercommand/query . . . . .	18-29
[SENSe:]FREQuency:MANualcommand/query . . . . .	18-30
[SENSe:]FREQuency:RESolutioncommand/query . . . . .	18-31
[SENSe:]FREQuency:RESolution:AUTOcommand/query . . . . .	18-34
[SENSe:]FREQuency:RESolution:AUTO:MCHangecommand/query . . . . .	18-36
[SENSe:]FREQuency:RESolution:AUTO:MINimumcommand/query . . . . .	18-37
[SENSe:]FREQuency:RESolution:OCTavecommand/query . . . . .	18-38
[SENSe:]FREQuency:SPANcommand/query . . . . .	18-40
[SENSe:]FREQuency:SPAN:FULLcommand . . . . .	18-43
[SENSe:]FREQuency:SPAN:LINKcommand/query . . . . .	18-45
[SENSe:]FREQuency:STARTcommand/query . . . . .	18-46
[SENSe:]FREQuency:STEP[:INCRement]command/query . . . . .	18-48

[SENSe:]FREQuency:STOPcommand/query . . . . .	18-49
[SENSe:]HISTogram:BINSommand/query . . . . .	18-50
[SENSe:]ORDer:MAXimumcommand/query. . . . .	18-51
[SENSe:]ORDer:RESolutioncommand/query. . . . .	18-52
[SENSe:]ORDer:RESolution:TRACkcommand/query . . . . .	18-53
[SENSe:]ORDer:RPM:MAXimumommand/query . . . . .	18-54
[SENSe:]ORDer:RPM:MINimumcommand/query . . . . .	18-55
[SENSe:]ORDer:TRACk[1 2 3 4 5]command/query . . . . .	18-56
[SENSe:]ORDer:TRACk[1 2 3 4 5]:STATecommand/query . . . . .	18-57
[SENSe:]REFErencecommand/query . . . . .	18-58
[SENSe:]REJect:STATecommand/query . . . . .	18-59
[SENSe:]SWEep:DIRectioncommand/query . . . . .	18-60
[SENSe:]SWEep:DWELlommand/query . . . . .	18-61
[SENSe:]SWEep:MODEcommand/query. . . . .	18-62
[SENSe:]SWEep:OVERlapcommand/query . . . . .	18-63
[SENSe:]SWEep:SPACingcommand/query . . . . .	18-65
[SENSe:]SWEep:STIMEcommand/query . . . . .	18-66
[SENSe:]SWEep:TIMEcommand/query . . . . .	18-67
[SENSe:]TCAPture:ABORTommand . . . . .	18-68
[SENSe:]TCAPture:DELetecommand . . . . .	18-69
[SENSe:]TCAPture:FILEommand/query . . . . .	18-70
[SENSe:]TCAPture[:IMMediate]command. . . . .	18-71
[SENSe:]TCAPture:LENGthcommand/query. . . . .	18-72
[SENSe:]TCAPture:MALLocatecommand . . . . .	18-74
[SENSe:]TCAPture:STARt[1 2 3 4]command/query . . . . .	18-75
[SENSe:]TCAPture:STOP[1 2 3 4]command/query . . . . .	18-76
[SENSe:]TCAPture:TACHometer:RPM:MAXimumommand/query. . . . .	18-77
[SENSe:]TCAPture:TACHometer[:STATE]command/query. . . . .	18-78
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:AUTOommand/query . . . . .	18-79
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:AUTO:DIRectioncommand/query . . . . .	18-81
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:UNIT:USER:LABelcommand/query . . . . .	18-82
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:UNIT:USER:SFACTorcommand/query . . . . .	18-83
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:UNIT:USER[:STATE]command/query . . . . .	18-84
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE:UNIT:XDCR:LABelcommand/query . . . . .	18-85
[SENSe:]VOLTage[1 2 3 4][:DC]:RANGE[:UPPER]command/query. . . . .	18-86
[SENSe:]WINDow[1 2 3 4]:EXponentialcommand/query . . . . .	18-87
[SENSe:]WINDow[1 2 3 4]:FORCecommand/query . . . . .	18-88
[SENSe:]WINDow[1 2 3 4]:ORDer:DCcommand/query . . . . .	18-89
[SENSe:]WINDow[1 2 3 4][:TYPE]command/query . . . . .	18-90
CHAPTER 19: SOURce	
SOURce:BURStcommand/query . . . . .	19-2
SOURce:FREQuency[:CW]command/query . . . . .	19-3
SOURce:FREQuency:FIXedcommand/query. . . . .	19-4
SOURce:FUNCTion[:SHAPE]command/query . . . . .	19-5
SOURce:USER:CAPTurecommand/query . . . . .	19-6
SOURce:USER[:REGister]ommand/query . . . . .	19-7
SOURce:USER:REPEatcommand/query . . . . .	19-8
SOURce:VOLTage[:LEVel]:AUTOcommand/query. . . . .	19-9
SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]command/query . . . . .	19-10
SOURce:VOLTage[:LEVel][:IMMediate]:OFFSetcommand/query . . . . .	19-11
SOURce:VOLTage[:LEVel]:REFerencecommand/query . . . . .	19-12
SOURce:VOLTage[:LEVel]:REFerence:CHANnelcommand/query. . . . .	19-13
SOURce:VOLTage[:LEVel]:REFerence:TOLerancecommand/query . . . . .	19-14
SOURce:VOLTage:LIMit[:AMPLitude]command/query . . . . .	19-15

SOURCE:VOLTage:LIMit:INPutcommand/query . . . . .	19-16
SOURCE:VOLTage:SLEWcommand/query. . . . .	19-17
CHAPTER 20: STATUs	
STATus:DEVice:CONDition?query . . . . .	20-2
STATus:DEVice:ENABlecommand/query . . . . .	20-3
STATus:DEVice[:EVENT]?query . . . . .	20-4
STATus:DEVice:NTRansitioncommand/query. . . . .	20-5
STATus:DEVice:PTRansitioncommand/query . . . . .	20-6
STATus:OPERation:CONDition?query. . . . .	20-7
STATus:OPERation:ENABlecommand/query . . . . .	20-8
STATus:OPERation[:EVENT]?query . . . . .	20-9
STATus:OPERation:NTRansitioncommand/query . . . . .	20-10
STATus:OPERation:PTRansitioncommand/query . . . . .	20-11
STATus:PRESetcommand . . . . .	20-12
STATus:QUEStionable:CONDition?query . . . . .	20-13
STATus:QUEStionable:ENABlecommand/query . . . . .	20-14
STATus:QUEStionable[:EVENT]?query . . . . .	20-15
STATus:QUEStionable:LIMit:CONDition?query . . . . .	20-16
STATus:QUEStionable:LIMit:ENABlecommand/query . . . . .	20-17
STATus:QUEStionable:LIMit[:EVENT]?query . . . . .	20-18
STATus:QUEStionable:LIMit:NTRansitioncommand/query. . . . .	20-19
STATus:QUEStionable:LIMit:PTRansitioncommand/query . . . . .	20-20
STATus:QUEStionable:NTRansitioncommand/query . . . . .	20-21
STATus:QUEStionable:PTRansitioncommand/query . . . . .	20-22
STATus:QUEStionable:VOLTage:CONDition?query . . . . .	20-23
STATus:QUEStionable:VOLTage:ENABlecommand/query. . . . .	20-24
STATus:QUEStionable:VOLTage[:EVENT]?query . . . . .	20-25
STATus:QUEStionable:VOLTage:NTRansitioncommand/query . . . . .	20-26
STATus:QUEStionable:VOLTage:PTRansitioncommand/query. . . . .	20-27
STATus:USER:ENABlecommand/query . . . . .	20-28
STATus:USER[:EVENT]?query . . . . .	20-29
STATus:USER:PULSecommand . . . . .	20-30
CHAPTER 21: SYSTEm	
SYSTem:BEEPer[:IMMediate]command. . . . .	21-2
SYSTem:BEEPer:STATecommand/query . . . . .	21-3
SYSTem:COMMunicate:GPIB[:SELF]:ADDReSScommand/query . . . . .	21-4
SYSTem:COMMunicate:SERial[:RECeive]:BAUDcommand/query . . . . .	21-5
SYSTem:COMMunicate:SERial[:RECeive]:BITScommand/query . . . . .	21-6
SYSTem:COMMunicate:SERial[:RECeive]:PACEcommand/query . . . . .	21-7
SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECKcommand/query. . . . .	21-8
SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]command/query. . . . .	21-9
SYSTem:COMMunicate:SERial[:RECeive]:SBITScommand/query . . . . .	21-10
SYSTem:COMMunicate:SERial:TRANsmiT:PACEcommand/query. . . . .	21-11
SYSTem:DATEcommand/query . . . . .	21-12
SYSTem:ERRor?query . . . . .	21-13
SYSTem:FAN[:STATe]command/query . . . . .	21-14
SYSTem:FLOG:CLEArcommand . . . . .	21-15
SYSTem:KEYcommand/query . . . . .	21-16
SYSTem:KLOCKcommand/query. . . . .	21-21
SYSTem:POWER:SOURce?query . . . . .	21-22
SYSTem:POWER:STATecommand/query. . . . .	21-23
SYSTem:PRESetcommand . . . . .	21-24
SYSTem:SETcommand/query . . . . .	21-25
SYSTem:TIMEcommand/query. . . . .	21-26

SYSTem:VERSion?query . . . . .	21-27
CHAPTER 22: TEST	
TEST:LOG:CLEArcommand . . . . .	22-2
TEST:LONGcommand . . . . .	22-3
TEST:LONG:RESult?query . . . . .	22-4
CHAPTER 23: TRACe	
TRACe[:DATA]command/query . . . . .	23-2
TRACe:WATerfall[:DATA]command/query . . . . .	23-5
TRACe:X[:DATA]?query . . . . .	23-7
TRACe:X:UNIT?query . . . . .	23-8
TRACe:Z[:DATA]query . . . . .	23-9
TRACe:Z:UNIT?query . . . . .	23-10
CHAPTER 24: TRIGger	
TRIGger:EXTernal:FILTer[:LPAS][:STATe]command/query . . . . .	24-2
TRIGger:EXTernal:LEVelcommand/query . . . . .	24-3
TRIGger:EXTernal:RANGecommand/query . . . . .	24-4
TRIGger[:IMMediate]command . . . . .	24-5
TRIGger:LEVelcommand/query . . . . .	24-6
TRIGger:SLOPecommand/query . . . . .	24-7
TRIGger:SOURcecommand/query . . . . .	24-8
TRIGger:STARt[1 2 3 4]command/query . . . . .	24-10
TRIGger:TACHometer:HOLDoffcommand/query . . . . .	24-12
TRIGger:TACHometer:LEVelcommand/query . . . . .	24-13
TRIGger:TACHometer:PCOuntcommand/query . . . . .	24-14
TRIGger:TACHometer:RANGecommand/query . . . . .	24-15
TRIGger:TACHometer[:RPM]?query . . . . .	24-16
TRIGger:TACHometer:SLOPecommand/query . . . . .	24-17
<b>APPENDIX A: Agilent 35670A Command Summary</b>	
Command List . . . . .	A-1
<b>APPENDIX B: Cross-Reference from Front-Panel Keys to GPIB Commands</b>	
Measurement Group . . . . .	B-2
Display Group . . . . .	B-9
Marker Group . . . . .	B-18
System Group . . . . .	B-21

**APPENDIX C: Error Messages**

Command Errors . . . . .	C-2
Execution Errors . . . . .	C-3
Device-Specific Errors . . . . .	C-10
Query Errors . . . . .	C-11

**APPENDIX D: Instrument Modes**

**APPENDIX E: Determining Units**

**APPENDIX F: Example Programs**

**INDEX**





## GPIB Programming with the Agilent 35670A

---



# GPIB Programming with the Agilent 35670A

---

## In This Book

This is GPIB Programming with the Agilent 35670A. It contains the command syntax, structure and a detailed description of each GPIB command available for the Agilent 35670A. In addition, it contains instrument-specific information not available in the GPIB Programmer's Guide.

For an introduction to GPIB programming, read the GPIB Programmer's Guide. It is intended for people not familiar with GPIB programming or remote control of an instrument. The book introduces the basic concepts of GPIB programming and describes the Standard Commands for Programmable Instruments (SCPI). In addition, it describes how to operate an instrument in an GPIB system and how to transfer data between an external controller and an instrument.

Chapter 1 presents GPIB programming information specific to the Agilent 35670A. It includes:

- How to setup the Agilent 35670A to an external controller and how to verify it works.
- A description of the Agilent 35670A's input, output and error queues.
- A listing of GPIB commands that require synchronization or passing control.
- A description of each of the Agilent 35670A's status registers.
- A listing of all SCPI commands.

Chapter 2 describes the conventions and syntax descriptions used in the command reference chapters, and a section on finding commands.

Chapters 3 - 24, the Command Reference, contain a detailed description of each GPIB command. The commands are organized alphabetically. A chapter introduction describes the SCPI subsystem and where it fits in the flow of control and measurement data within the Agilent 35670A.

The appendices, A - F, contain a variety of reference material:

- Appendix A provides a quick reference to the Agilent 35670A's GPIB command set.
- Appendix B provides a cross reference of the Agilent 35670A's hardkeys and softkeys and their equivalent GPIB commands.
- Appendix C provides a complete listing of the Agilent 35670A's error messages.
- Appendix D provides a list of valid GPIB commands for each of the major instrument modes.
- Appendix E explains how to determine the Y-axis units you send with certain commands.
- Appendix F lists some of the example programs for the Agilent 35670A that appear on the Agilent 35670A Example Programs Disk. The example programs demonstrate how to transfer data between the Agilent 35670A and an external controller.

Included in the documentation set is the "Agilent 35670A GPIB Commands: Quick Reference." This card provides quick and convenient access to command syntax and structure.

## GPIB Setup

This section contains a procedure for configuring the Agilent 35670A and an external controller in a simple GPIB system. Although an HP 9000 Series 340 computer is the controller used in the system, other computers that support an GPIB interface can also be used. If you are using one of those other computers, the configuration procedure can only be used as a general guide. You should consult your computer's documentation for more complete information.

This section also contains a procedure for verifying that commands can be sent over the GPIB. BASIC is used for the verification procedure's test program. If your computer uses some other language, the keywords and syntax for the test program may be different. You will need to write a similar program using your language's keywords and syntax.

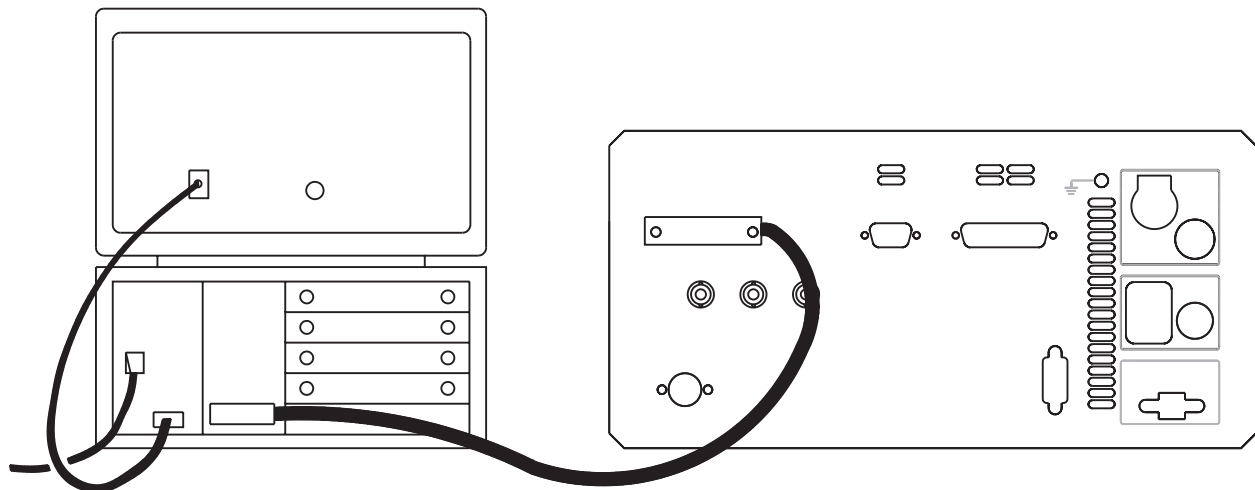
### Configuring the GPIB System

#### Equipment and Software

- Agilent 35670A Dynamic Signal Analyzer
- HP 9000 Series 340 computer
- Agilent 10833A, B, C, or D GPIB Cable
- BASIC

#### Procedure

Turn off the Agilent 35670A and the HP 9000 Series 340, then connect them with the GPIB cable as shown in figure 1-1.



3-1. GPIB Connections

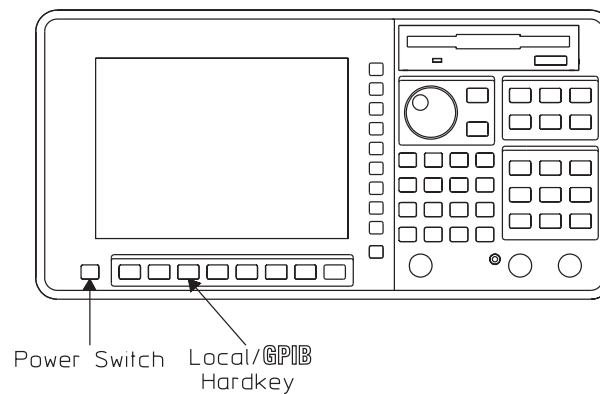
1. Turn on the HP 9000 Series 340. If necessary, load BASIC following the instructions in the computer's operating manual. Note that the following language extensions must be installed for the verification program to work:

- CRTA
- GPIB
- IO
- EDIT

Programs that are more complex than the verification program will probably require more language extensions. For a complete list of loaded language extensions, enter the following BASIC command into your computer:

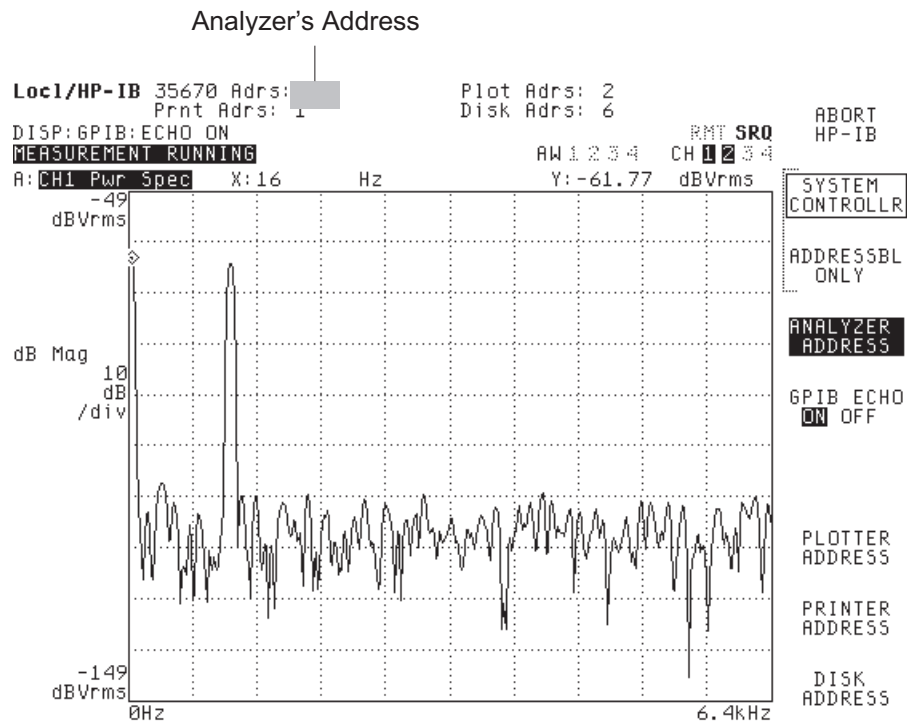
```
LIST BIN
```

Turn on the Agilent 35670A. When the softkey labels appear, press the [ Local/GPIB ] hardkey. See figure 1-2.



1-2. Agilent 35670A Front Panel

Verify that the analyzer's address is set to 11. The current address setting is displayed when you press the [ ANALYZER ADDRESS ] softkey (see figure 1-3). You can change the address by pressing [ ANALYZER ADDRESS ], then using the numeric keypad and the [ ENTER ] softkey to enter a new value. However, the instructions in the verification procedure assume that the analyzer address is set to 11.



3-3. Agilent 35670A Screen After Pressing Local/GPIB

Verify that the analyzer is set to the addressable-only mode. The softkey labels that appear when you press the [ Local/GPIB ] hardkey include [ SYSTEM CONTROLLR ] and [ ADDRESSBL ONLY ]. Only one of these two softkeys can be selected at a time, and the one that is selected will have a box around it. Press [ ADDRESSBL ONLY ] if it is not selected.

---

**Note** In any GPIB system there can be more than one device with controller capabilities. But at any given time, only one device on the bus can be designated as the system controller. See the GPIB Programmer's Guide for more information about controller capabilities.

---

### Quick Verification

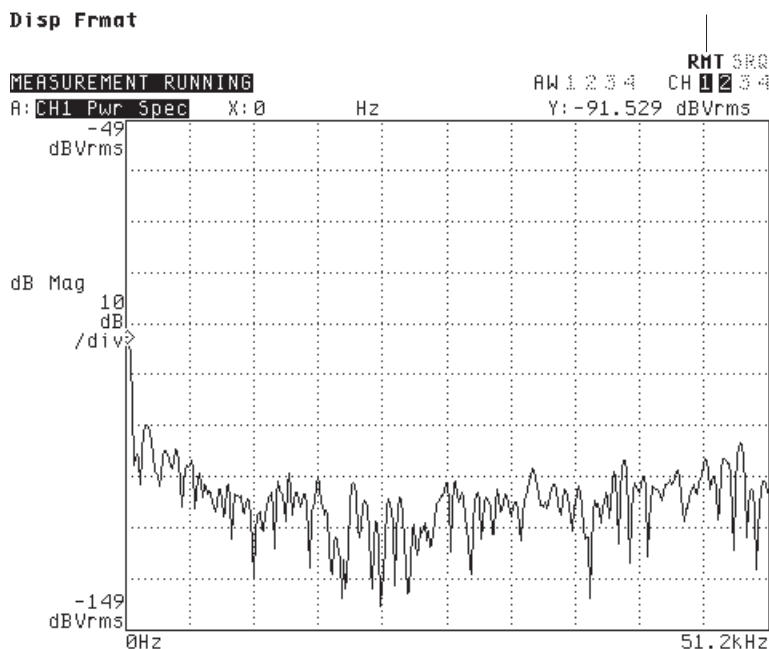
Having just completed all the steps in the preceding section, you are ready to verify that commands can be sent over the GPIB. In this quick verification, you are going to enter an BASIC keyword that should place the Agilent 35670A under remote control.

#### Procedure

Type the following on the computer:

```
REMOTE 711
```

then press the computer's ENTER key. The RMT indicator should appear highlighted at the top of the Agilent 35670A's screen (see figure 1-4). This tells you that the analyzer is under remote control of the computer.



3-4. RMT (Remote) Indicator

#### Remote Indicator

Now type the following on the computer:

```
LOCAL 711
```

then press the computer's ENTER key. The RMT indicator should become "ghosted." That is, the word is still readable, but no longer highlighted. This tells you that the analyzer has been returned to front-panel control.

## GPIB Setup

### Troubleshooting

If the RMT indicator does not perform as expected, check the following:

- Be sure that your GPIB cable connections are secure and that the cable is free of defects.
- Verify that the analyzer is in addressable-only mode and that its address is set to 11.
- Be sure you are using the required equipment and software.
- Be sure you have loaded all the required language extensions into the computer. (For a list of loaded extensions, enter the following into the computer: LIST BIN)

If everything seems to be in order, but the RMT indicator still doesn't perform as expected, follow the procedure in "Need Assistance?," that appears on the last page of this book.



## Verification Program

The quick verification procedure confirmed that the computer could talk to the analyzer. However, you must write a short program to confirm that the analyzer can talk to the computer. If you enter the program correctly, the computer displays the following statement when you run the program:

FREQUENCY SPAN IS: 51200 HZ

---

**Note**            The following procedure assumes that you have completed all the steps in “Configuring the GPIB System” using all the required equipment and software.

---

### Procedure

Enter the following program:

```

10  PRINTER IS 1
20  ASSIGN @Agilent35670A TO 711
30  ABORT 7
40  CLEAR @Agilent35670A
50  OUTPUT @Agilent35670A;"*RST"
60  OUTPUT @Agilent35670A;"SENS:FREQ:SPAN:FULL"
70  OUTPUT @Agilent35670A;"SENS:FREQ:SPAN?"
80  ENTER @Agilent35670A;A
90  PRINT "FREQUENCY SPAN IS:";A;"HZ"
100 END

```

See your computer and software documentation if you need help entering the program.

RUN the program. The program tells the analyzer to reset. It then tells the analyzer to select its widest frequency span. Finally, the program asks the analyzer to return the value of the widest span and has the computer display the returned value as follows:

FREQUENCY SPAN IS: 51200 HZ

### Troubleshooting

If the program does not run correctly, be sure you have entered the program exactly as listed. Then go back to “Quick Verification” for additional troubleshooting hints.

## How the Agilent 35670A Operates in an GPIB System

This section provides instrument-specific information for the operation of the Agilent 35670A in an GPIB system. For a general overview of how an analyzer operates in an GPIB system, see the GPIB Programmer's Guide .

### Controller Capabilities

The Agilent 35670A can be configured as an GPIB system controller or as an addressable-only GPIB device. To configure the analyzer, press the [ Local/GPIB ] key on the front panel. To configure the analyzer as the GPIB system controller press [SYSTEM CONTROLLER] which appears in the softkey menu. To configure the Agilent 35670A as an addressable-only device on the bus, press [ADDRESSBL ONLY].

Normally, the Agilent 35670A is not configured as the system controller unless it is the only controller on the bus. Such a setup would be likely if you only wanted to control printers or plotters with the analyzer. It might also be the case if you were using Instrument BASIC to control other test equipment.

When the analyzer is used with another controller on the bus, it is normally configured as an addressable-only GPIB device. In this configuration, when the analyzer is passed control it can function as the active controller. It can also function as a talker or listener.

### GPIB Commands That Require Passing Control to the Agilent 35670A

If there is more than one controller on the bus, the Agilent 35670A needs to have active control (instead of system control) to initiate operations such as plotting. The Agilent 35670A must be the active controller for the following GPIB commands:

- HCOpy[:IMMediate]
- HCOpy:ITEM:ALL[:IMMediate]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:GRATicule[:IMMediate]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe[:IMMediate]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:MARKer[:IMMediate]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:MARKer:REFerence[:IMMediate]
- MMEMory:COpy
- MMEMory:DELeTe
- MMEMory:INITialize
- MMEMory:MSIS
- MMEMory:MOVE

See the GPIB Programmer's Guide, for more information and examples about passing control to an analyzer.

## GPIB Interface Capabilities

The Agilent 35670A has the following interface capabilities, as defined by the IEEE 488.1 standard:

SH1	full Source handshake capability
AH1	full Acceptor handshake capability
T6	basic Talker, Serial Poll, no Talk Only, unaddress if MLA
TE0	no Extended Talker capability
L4	basic Listener, no Listen Only, unaddress if MTA
LE0	no Extended Listener capability
SR1	full Service Request capability
RL1	full Remote/Local capability
PP0*	Parallel Poll capability
DC1	full Device Clear capability
DT1	full Device Trigger capability
C1	System Controller capability
C2	send IFC and take charge Controller capability
C3	send REN Controller capability
C4*	respond to SRQ
C6*	send IFC, receive control, pass control, parallel poll, pass control to self
C10*	send IFC, receive control, pass control, parallel poll
C12	send IF messages, receive control, pass control
E2	tri-state drivers

\* only when an Instrument BASIC program is running

## **GPIB Queues in the Agilent 35670A**

Queues enhance the exchange of messages between the Agilent 35670A and other devices on the bus. The Agilent 35670 contains three queues.

- The input queue holds up to 128 bytes.
- The error queue temporarily stores up to 5 error messages.
- The output queue temporarily stores a single response message until it is read by a controller.

For additional information about the use of queues in exchanging messages between an analyzer and an external controller, see the GPIB Programmer's Guide .

## **Command Synchronization**

Device commands can be divided into two broad classes:

- sequential commands
- overlapped commands

Most device commands that you send to the analyzer are processed sequentially. A sequential command holds off the processing of subsequent commands until it has been completely processed. Some commands do not hold off the processing of subsequent commands. These are called overlapped commands and in many situations they require synchronization.

See the GPIB Programmer's Guide for more information and examples about command synchronization.

## Overlapped Commands in the Agilent 35670A

The Agilent 35670A has the following overlapped commands that require synchronization:

- CALCulate[1|2|3|4]:CFIT[:IMMEDIATE]
- CALCulate[1|2|3|4]:FEED
- CALCulate[1|2|3|4]:FORMat
- CALCulate[1|2|3|4]:SYNTHeSis[:IMMEDIATE]
- CALCulate[1|2|3|4]:UNIT:AMPLitude
- CALCulate[1|2|3|4]:UNIT:ANGLE
- CALCulate[1|2|3|4]:UNIT:DBReference
- CALCulate[1|2|3|4]:UNIT:DBReference:IMPedance
- CALCulate[1|2|3|4]:UNIT:DBReference:USER:LABel
- CALCulate[1|2|3|4]:UNIT:DBReference:USER:REFerence
- CALCulate[1|2|3|4]:UNIT:VOLTage
- CALCulate[1|2|3|4]:UNIT:X
- CALCulate[1|2|3|4]:UNIT:X:ORDER:FACTor
- CALCulate[1|2|3|4]:UNIT:X:USER:FREQuency:FACTor
- CALCulate[1|2|3|4]:UNIT:X:USER:FREQuency:LABel
- CALCulate[1|2|3|4]:UNIT:X:USER:TIME:FACTor
- CALCulate[1|2|3|4]:UNIT:X:USER:TIME:LABel
- HCOpy[:IMMEDIATE]
- HCOpy:ITEM:ALL[:IMMEDIATE]
- INITiate[:IMMEDIATE]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:GRATicule[:IMMEDIATE]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe[:IMMEDIATE]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:MARKer[:IMMEDIATE]
- HCOpy:ITEM[:WINDOW[1|2|3|4]]:TRACe:MARKer:REFerence[:IMMEDIATE]
- MMEMory:COpy
- MMEMory:DELeTe
- MMEMory:INITialize
- MMEMory:REName
- [SENSe:]TCAPture[:IMMEDIATE]

## **GPIB Commands That Transfer Mixed (ASCII/REAL) Data**

The FORMat:DATA command selects the type of data and the type of data encoding that is used to transfer large blocks of numeric data between the analyzer and an external controller. Block data that contains mixed data—both REAL numbers and ASCII characters—ignore the setting of the FORMat:DATA command. These blocks always transfer as either definite or indefinite length block data. The following commands transfer blocks of mixed data:

- CALCulate:MATH:DATA
- PROGram:EXPLicit:DEFine
- PROGram[:SELEcted]:DEFine
- SYSTem:SET

## **Determining Units**

You can determine the units associated with a set value by sending the unit parameter with the command's query form. For example, send

```
SENSE:VOLTAGE:RANGE? UNIT
```

to determine the units associated with the value of the input range.

## The Agilent 35670A's Status Registers

### Register Summary

The Agilent 35670A uses eight register sets to keep track of instrument status:

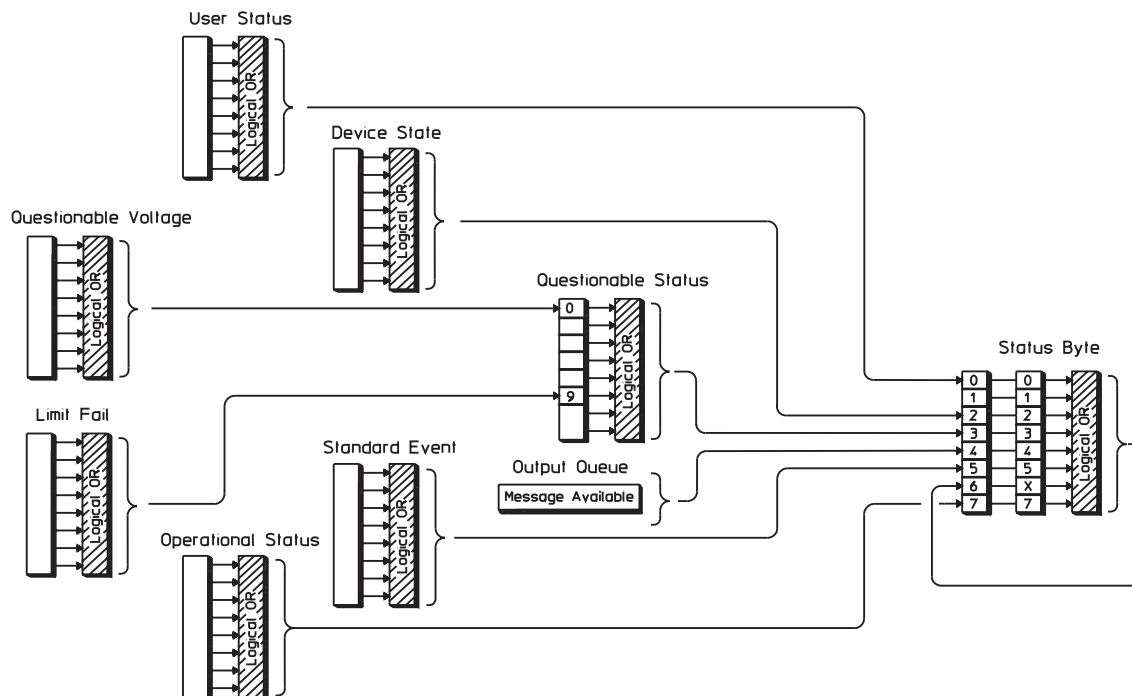
- Status Byte
- Device State
- Limit Fail
- Questionable Status
- Questionable Voltage
- Standard Event
- Operation Status
- User Status

Their reporting structure is summarized in figure 1-5. They are described in greater detail in the following sections.

---

**Note** Register bits not explicitly presented in the following sections are not used by the Agilent 35670A. A query to one of these bits returns a value of 0.

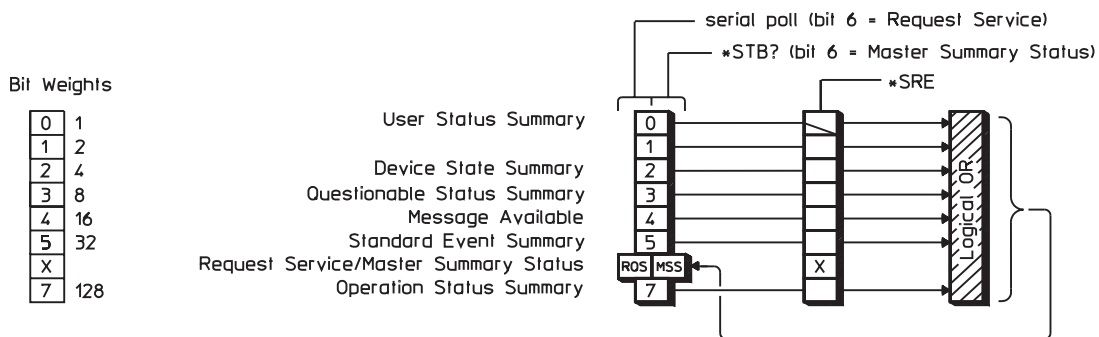
---



1-5. Agilent 35670A Register Sets

## Status Byte Register Set

The Status Byte register set summarizes the states of the other register sets and monitors the analyzer's output queue. It is also responsible for generating service requests. See figure 1-6.



1-6. The Status Byte Register Set

The Status Byte register set contains only two registers: the Status Byte register and the Service Request enable register. The Status Byte register behaves like a condition register for all bits except bit 6. The Service Request enable register behaves like a standard enable register except that bit 6 is always set to 0.

For more information about generating service requests, see "How to Use Registers" in the GPIB Programmer's Guide.



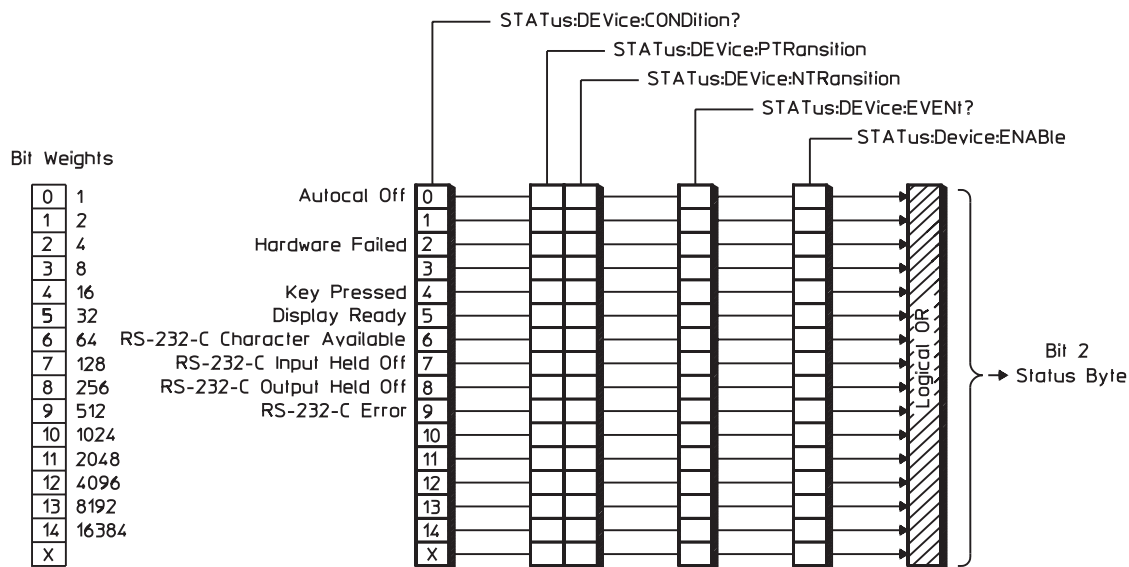
Bits in the Status Byte register are set to 1 under the following conditions:

- User Status Summary (bit 0) is set to 1 when one or more enabled bits in the User Status event register are set to 1.
- Device State Summary (bit 2) is set to 1 when one or more enabled bits in the Device State event register are set to 1.
- Questionable Status Summary (bit 3) is set to 1 when one or more enabled bits in the Questionable Status event register are set to 1.
- Message Available (bit 4) is set to 1 when the output queue contains a response message.
- Standard Event Summary (bit 5) is set to 1 when one or more enabled bits in the Standard Event event register are set to 1.
- Master Summary Status (bit 6, when read by \*STB) is set to 1 when one or more enabled bits in the Status Byte register are set to 1.
- Request Service (bit 6, when read by serial poll) is set to 1 by the service request process (see “How to Use Registers” in the *GPIB Programmer's Guide*).
- Operation Status Summary (bit 7) is set to 1 when one or more enabled bits in the Operation Status event register are set to 1.

Figure 1-6 also shows the commands you use to read and write the Status Byte registers. See chapter 20 for more information about these commands.

## Device State Register Set

The Device State register set monitors the states of eight device-specific parameters. See figure 1-7.



1-7. The Device State Register Set

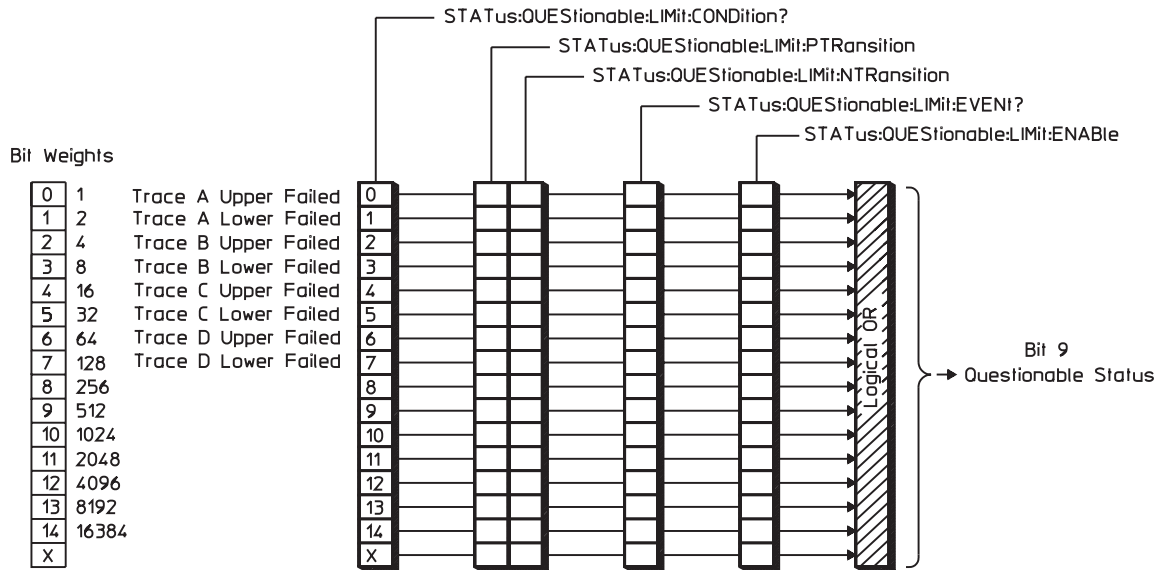
Bits in the Device State condition register are set to 1 under the following conditions:

- Autocal Off (bit 0) is set to 1 when the analyzer's autocalibration function is disabled (CAL:AUTO OFF).
- Hardware Failed (bit 2) is set to 1 when the analyzer detects a failure in its own hardware.
- Key Pressed (bit 4) is set to 1 when one of the front panel keys is pressed. This is an event. The condition register will always return 0 for this bit.
- Display Ready (bit 5) is set to 1 when measurement results are available. This is an event. The condition register will always return 0 for this bit.
- RS-232-C Character Available (bit 6) is set to 1 when a character is in the input buffer.
- RS-232-C Input Held Off (bit 7) is set to 1 when input is held off due to handshake protocol conditions.
- RS-232-C Output Held Off (bit 8) is set to 1 when output is held off due to handshake protocol conditions.
- RS-232-C Error (bit 9) is set to 1 when a framing error, overrun error, parity error, or break is detected.

Figure 1-7 also shows the commands you use to read and write the Device State registers. See chapter 20 for more information about these commands.

## Limit Fail Register Set

The Limit Fail register set monitors limit test results for all traces. See figure 1-8.



1-8. The Limit Fail Register Set

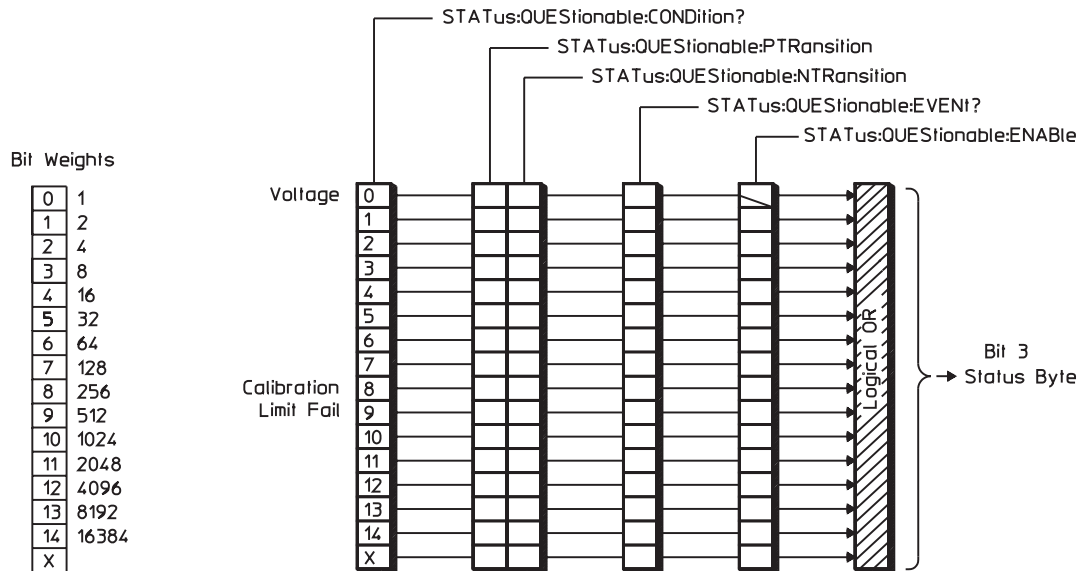
Bits in the Limit Fail condition register are set to 1 under the following conditions:

- Trace A Upper Failed (bit 0) is set to 1 when limit testing is enabled and any point on trace A exceeds its upper limit.
- Trace A Lower Failed (bit 1) is set to 1 when limit testing is enabled and any point on trace A falls below its lower limit.
- Trace B Upper Failed (bit 2) is set to 1 when limit testing is enabled and any point on trace B exceeds its upper limit.
- Trace B Lower Failed (bit 3) is set to 1 when limit testing is enabled and any point on trace B falls below its lower limit.
- Trace C Upper Failed (bit 4) is set to 1 when limit testing is enabled and any point on trace C exceeds its upper limit.
- Trace C Lower Failed (bit 5) is set to 1 when limit testing is enabled and any point on trace C falls below its lower limit.
- Trace D Upper Failed (bit 6) is set to 1 when limit testing is enabled and any point on trace D exceeds its upper limit.
- Trace D Lower Failed (bit 7) is set to 1 when limit testing is enabled and any point on trace D falls below its lower limit.

Figure 1-8 also shows the commands you use to read and write the Limit Fail registers. See chapter 20 for more information about these commands.

## Questionable Status Register Set

The Questionable Status register set monitors conditions that affect the quality of measurement data. See figure 1-9.



1-9. The Questionable Status Register Set

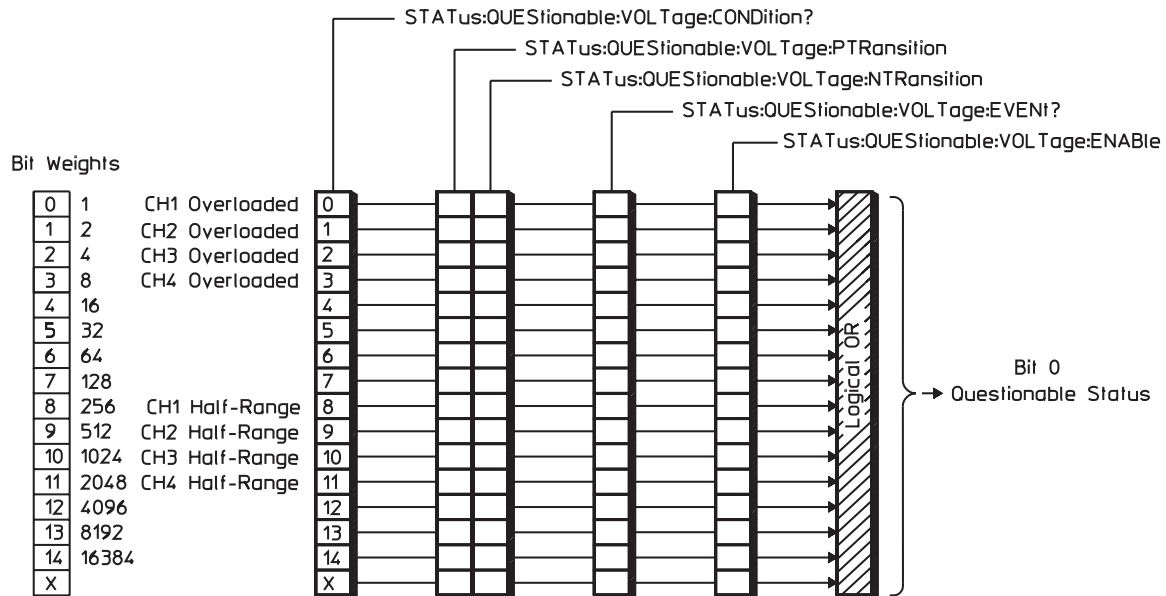
Bits in the Questionable Status condition register are set to 1 under the following conditions:

- Voltage (bit 0) is set to 1 when one or more enabled bits in the Questionable Voltage event register are set to 1.
- Calibration (bit 8) is set to 1 when the last self-calibration attempted by the analyzer failed.
- Limit Fail (bit 9) is set to 1 when one or more enabled bits in the Limit Fail event register are set to 1.

Figure 1-9 also shows the commands you use to read and write the Questionable Status registers. See chapter 20 for more information about these commands.

## Questionable Voltage Register Set

The Questionable Voltage register set monitors conditions that affect the amplitude accuracy of measurement data. See figure 1-10.



1-10. The Questionable Voltage Register Set

Bits in the Questionable Voltage condition register are set to 1 under the following conditions:

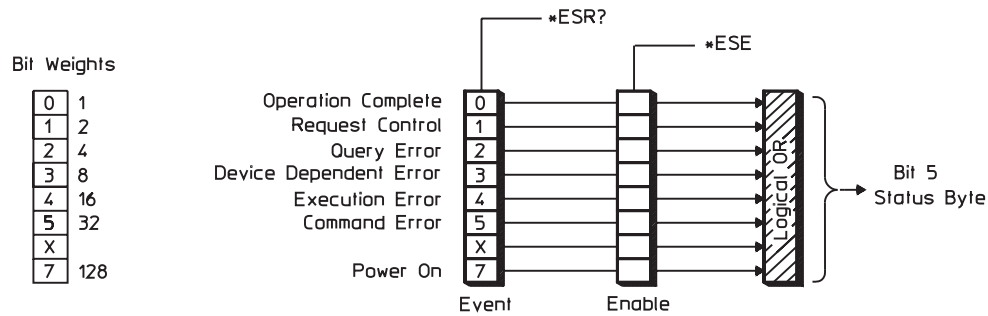
- Channel 1 Overload (bit 0) is set to 1 when any input signal exceeds the current channel 1 input range.
- Channel 2 Overload (bit 1) is set to 1 when any input signal exceeds the current channel 2 input range.
- Channel 3 Overload (bit 2) in Option AY6 only is set to 1 when any input signal exceeds the current channel 3 input range.
- Channel 4 Overload (bit 3) in Option AY6 only is set to 1 when any input signal exceeds the current channel 4 input range.
- Channel 1 Input Half-Range (bit 8) is set to 1 when any input signal is larger than half the current channel 1 input range.
- Channel 2 Input Half-Range (bit 9) is set to 1 when any input signal is larger than half the current channel 2 input range.
- Channel 3 Input Half-Range (bit 10) in Option AY6 only, is set to 1 when any input signal is larger than half the current channel 3 input range.
- Channel 4 Input Half-Range (bit 11) in Option AY6 only, is set to 1 when any input signal is larger than half the current channel 4 input range.

Figure 1-10 also shows the commands you use to read and write the Questionable Voltage registers. See chapter 20 for more information about these commands.

### Standard Event Register Set

The Standard Event register set monitors GPIB errors and synchronization conditions. See

figure 1-11.



1-11. The Standard Event Register Set

The Standard Event register set contains only two registers: the Standard Event event register and the Standard Event enable register. The Standard Event event register is similar to other event registers, but behaves like a positive transition register with all bits set to 1. The Standard Event enable register is the same as other enable registers.

For more information about the behavior of the Standard Event register set, see the GPIB Programmer's Guide .

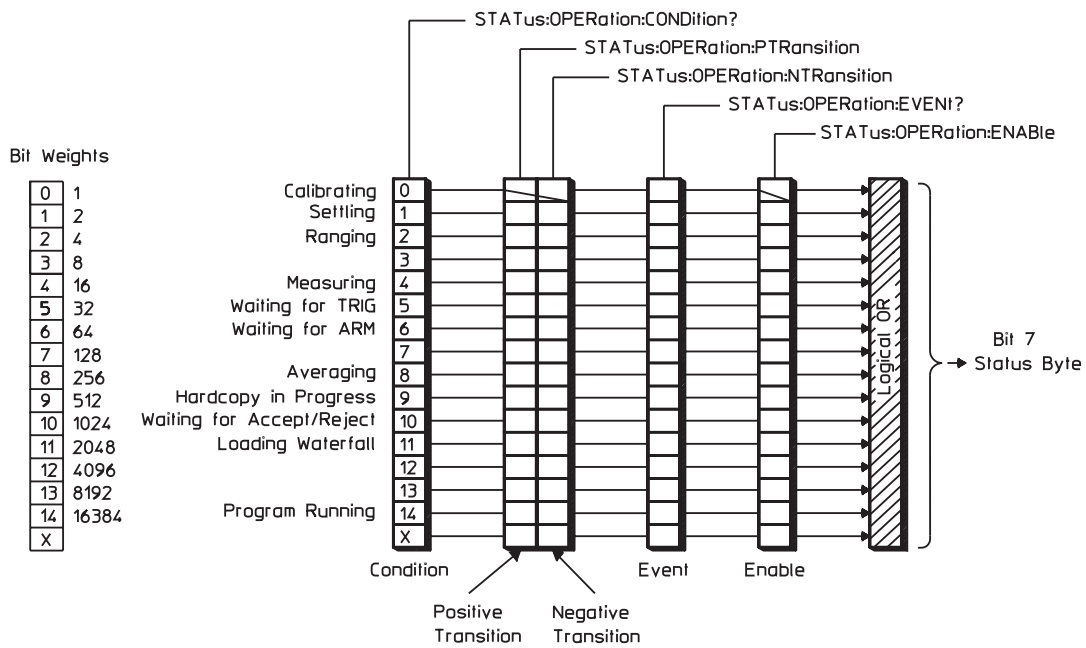
Bits in the Standard Event event register are set to 1 under the following conditions:

- Operation Complete (bit 0) is set to one when the following two events occur (in the order listed):
  - You send the \*OPC command to the analyzer.
  - The analyzer completes all pending overlapped commands (see “Command Synchronization” earlier in this chapter).
- Request Control (bit 1) is set to 1 when both of the following conditions are true:
  - The analyzer is configured as an addressable-only GPIB device (see “Controller Capabilities” earlier in this chapter).
  - The analyzer is instructed to do something (such as plotting or printing) that requires it to take control of the bus.
- Query Error (bit 2) is set to 1 when the command parser detects a query error.
- Device Dependent Error (bit 3) is set to 1 when the command parser detects a device-dependent error.
- Execution Error (bit 4) is set to 1 when the command parser detects an execution error.
- Command Error (bit 5) is set to 1 when the command parser detects a command error.
- Power On (bit 7) is set to 1 when you turn on the analyzer.

Figure 1-11 also shows the commands you use to read and write the Standard Event registers. See chapter 20 for more information about these commands.

## Operation Status Register Set

The Operation Status register set monitors conditions in the analyzer's measurement process, disk operations, and printing/plotting operations. It also monitors the state of current Instrument BASIC program. See figure 1-12.



1-12. The Operation Status Register Set



Bits in the Operation Status condition register are set to 1 under the following conditions:

- Calibrating (bit 0) is set to 1 while the self-calibration routine is running.
- Settling (bit 1) is set to 1 while the measurement hardware is settling.
- Ranging (bit 2) is set to 1 while the input range is changing.
- Measuring (bit 4) is set to 1 while the analyzer is collecting data for a measurement.
- Waiting for TRIG (bit 5) is set to 1 when the analyzer is ready to accept a trigger signal from one of the trigger sources. (If a trigger signal is sent before this bit is set, the signal is ignored.)
- Waiting for ARM (bit 6) is set to 1 when both of the following conditions are true:
  - Manual arming is selected.
  - The analyzer is ready to be armed.

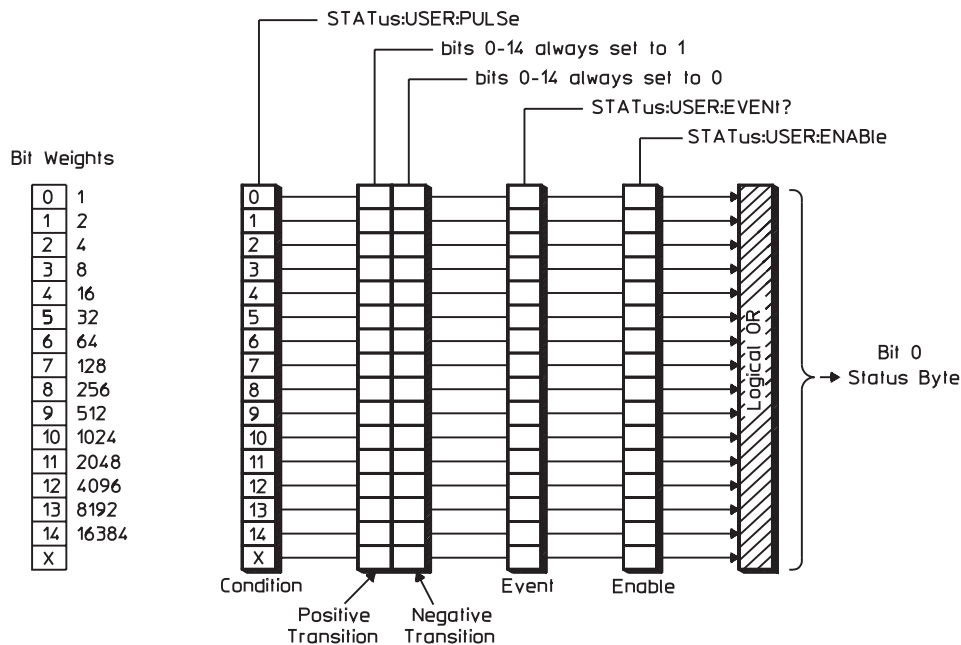
(If you send the ARM:IMM command before this bit is set, the command is ignored.)
- Averaging (bit 8) is set to 1 while the analyzer is averaging measurement data. If averaging is disabled ([SENSe:]AVERAge[:STATe] OFF) this bit is set to 1 whenever the Measuring bit (bit 4) is set to 1 during data collection.
- Hardcopy In Progress (bit 9) is set to 1 while the analyzer is performing a print or plot operation.
- Waiting for Accept/Reject (bit 10) is set to 1 while the analyzer is waiting for a response during preview averaging ([SENSe:]AVERAge:PREView MANual).
- Loading Waterfall (bit 11) is set to 1 while the analyzer is collecting the specified number of traces for a waterfall display.
- Program Running (bit 14) is set to 1 while the current Instrument BASIC program is running.

Figure 18-1 under the [SENSe:]AVERAge[:STATe] command illustrates the transition of the bits in the Operation Status condition register.

Figure 1-12 also shows the commands you use to read and write the Operation Status registers. See chapter 20 for more information about these commands.

## User Status Register Set

The User Status register set detects STATus:USER:PULSe commands. See figure 1-13.



1-13. The User Status Register Set

The User Status register set conforms to the general status register model (described at the beginning of this chapter) with the following exceptions:

- You can write (but not read) the condition register.
- You cannot write or read the transition registers.
- Bits in the positive transition register are always set to 1.
- Bits in the negative transition register are always set to 0.
- Bit 15 is not available. It is always set to 0.

Bits in the User Status condition register are normally set to 0, but are set to 1 (briefly) when you send a STAT:USER:PULS command. If you send STAT:USER:PULS 32, bit 5 of the condition register is pulsed high ( $2^5 = 32$ ).

Figure 1-13 also shows the commands you use to read or write the User Status registers. See chapter 20 for more information about these commands.

### Agilent 35670A Register Set Summary

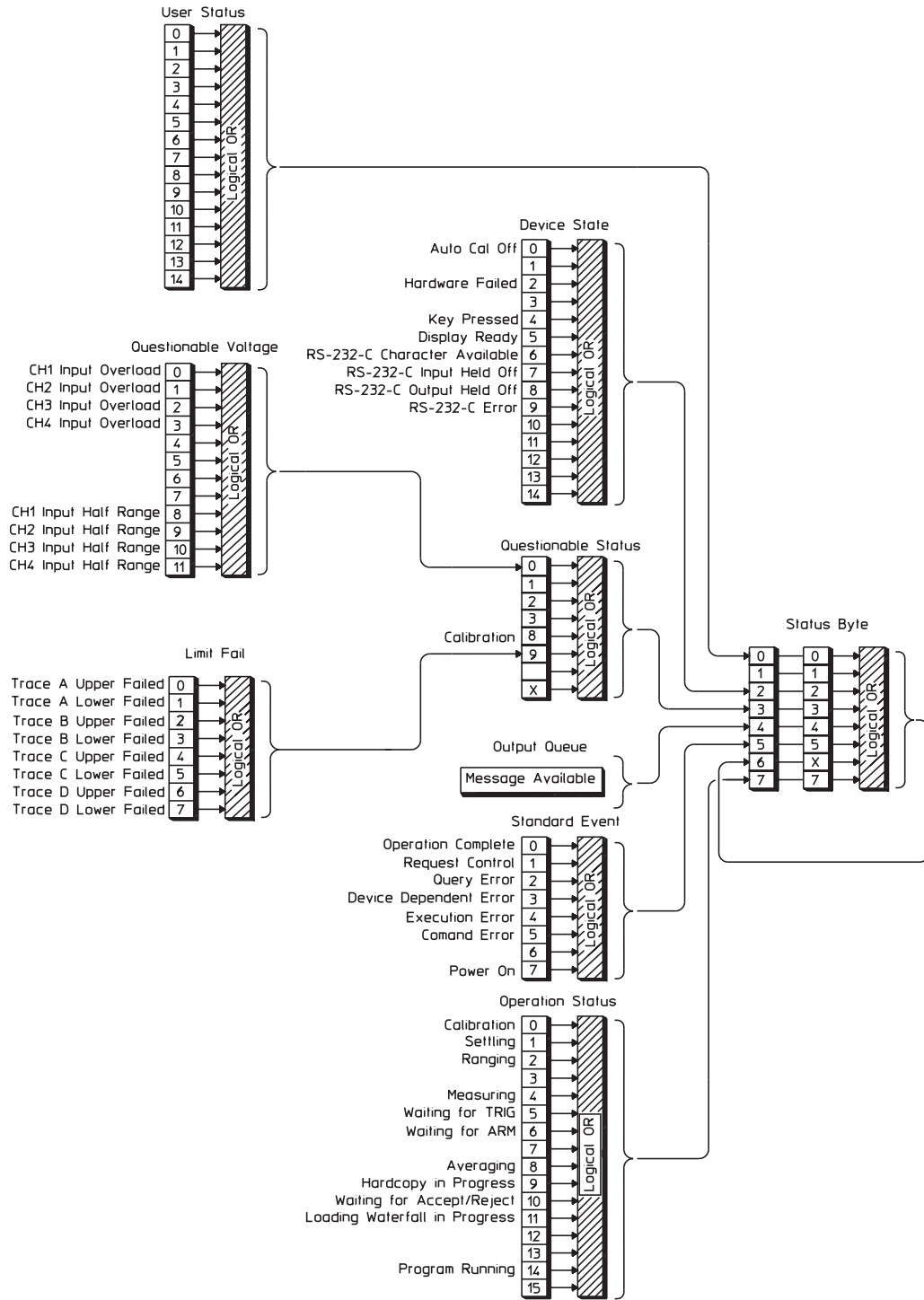


Figure 1-14. Agilent35670A Register Set Summary

## SCPI Compliance Information

Many of the Agilent 35670A's GPIB commands comply to SCPI. The attribute summary in the command reference section identifies these commands as follows:

- Confirmed commands which comply to SCPI 1992.
- Approved commands which will be added to SCPI 1993.
- Instrument-specific commands which do not comply to SCPI.

Use the `SYSTEM:VERSion?` query to determine the SCPI version to which your analyzer complies.

To enter the query in BASIC, you simply type

```
OUTPUT @711;"SYSTEM:VERSION?"
```

```
ENTER @711;version
```

The analyzer returns a value that looks like this:

```
1992.0
```

where 1992 is the year-version and 0 is the revision number for that year.

**Confirmed Commands**

\*CAL?  
 \*CLS  
 \*ESE  
 \*ESR?  
 \*IDN?  
 \*OPC  
 \*OPT?  
 \*PCB-?  
 \*PSC  
 \*RST  
 \*SRE  
 \*STB?  
 \*TRG

ABORt

ARM[:IMMediate]  
 ARM:SOURce

CALCulate[1|2]:DATA?  
 CALCulate[1|2]:FEED  
 CALCulate[1|2]:FORMat  
 CALCulate[1|2]:LIMit:FAIL?  
 CALCulate[1|2]:LIMit:STATe  
 CALCulate[1|2]:MATH:STATe  
 CALCulate[1|2]:MATH[:EXPRession[1|2|3|4|5]]  
 CALCulate[1|2]:UNIT:ANGLE

CALibration[:ALL]?  
 CALibration:AUTO

DISPlay:ANNotation[:ALL]  
 DISPlay:CONTents  
 DISPlay:ENABle  
 DISPlay:FORMat  
 DISPlay:PROGram[:MODE]  
 DISPlay[:WINDow[1|2]]:TRACe:GRATicule:GRID[:STATe]  
 DISPlay[:WINDow[1|2]]:TRACe:X:SPACing  
 DISPlay[:WINDow[1|2]]:TRACe:X[:SCALe]:LEFT  
 DISPlay[:WINDow[1|2]]:TRACe:X[:SCALe]:RIGHT  
 DISPlay[:WINDow[1|2]]:TRACe:Y[:SCALe]:AUTO  
 DISPlay[:WINDow[1|2]]:TRACe:Y[:SCALe]:BOTTom  
 DISPlay[:WINDow[1|2]]:TRACe:Y[:SCALe]:PDIVision  
 DISPlay[:WINDow[1|2]]:TRACe:Y[:SCALe]:TOP  
 DISPlay[:WINDow[1|2]]:TRACe:Y:SPACing

## SCPI Compliance Information

FORMat[:DATA]

INITiate:CONTinuous

INITiate[:IMMediate]

INPut[1|2]:BIAS[:STATe]

INPut[1|2]:COUPling

INPut[1|2]:FILTer:AWEighting[:STATe]

INPut[1|2]:FILTer[:LPASs][:STATe]

INPut[1|2]:LOW

INPut[1|2][:STATe]

INSTrument:NSElect

INSTrument[:SElect]

MEMory:CATalog[:ALL]?

MEMory:DElete:ALL

MEMory:DElete[:NAME]-?

MEMory:FREE[:ALL]?

MMEMory:COpy-?

MMEMory:DElete-?

MMEMory:INITialize-?

MMEMory:LOAD:STATe-?

MMEMory:LOAD:TRACe-?

MMEMory:MOVE-?

MMEMory:MSIS

MMEMory:STORe:STATe-?

MMEMory:STORe:TRACe-?

OUTPut[:STATe]

PROGram:EXPLicit:DEFine

PROGram[:SElected]:DEFine

PROGram[:SElected]:DElete:ALL

PROGram[:SElected]:DElete[:SElected]

PROGram[:SElected]:MALLocate

PROGram[:SElected]:NAME

PROGram[:SElected]:NUMBer

PROGram[:SElected]:STATe

PROGram[:SElected]:STRing

SOURce:FREQuency[:CW]  
 SOURce:FREQuency:FIXed  
 SOURce:FUNcTion[:SHApe]  
 SOURce:VOLTagE:LIMit[:AMPLitude]  
 SOURce:VOLTagE[:LEVel][:IMMediate][:AMPLitude]  
 SOURce:VOLTagE:SLEW

[SENSe:]AVERAge:COUNT  
 [SENSe:]AVERAge:TCONtrol  
 [SENSe:]AVERAge:TYPE  
 [SENSe:]AVERAge[:STATe]  
 [SENSe:]FREQuency:CENTer  
 [SENSe:]FREQuency:MANual  
 [SENSe:]FREQuency:RESolution  
 [SENSe:]FREQuency:RESolution:AUTO  
 [SENSe:]FREQuency:SPAN  
 [SENSe:]FREQuency:SPAN:FULL  
 [SENSe:]FREQuency:SPAN:LINK  
 [SENSe:]FREQuency:STARt  
 [SENSe:]FREQuency:STEP[:INCRement]  
 [SENSe:]FREQuency:STOP  
 [SENSe:]SWEep:DIRection  
 [SENSe:]SWEep:DWELl  
 [SENSe:]SWEep:MODE  
 [SENSe:]SWEep:SPACing  
 [SENSe:]SWEep:TIME  
 [SENSe:]VOLTagE[1|2]:RANGe:AUTO  
 [SENSe:]VOLTagE[1|2]:RANGe[:UPPer]  
 [SENSe:]WINDow[1|2]:EXPOntial  
 [SENSe:]WINDow[1|2]:FORCe  
 [SENSe:]WINDow[1|2][:TYPE]

STATus:OPERation:CONDition?  
 STATus:OPERation:ENABle  
 STATus:OPERation:NTRansition  
 STATus:OPERation:PTRansition  
 STATus:OPERation[:EVENT]?  
 STATus:PRESet  
 STATus:QUEStionable:CONDition?  
 STATus:QUEStionable:ENABle  
 STATus:QUEStionable:NTRansition  
 STATus:QUEStionable:PTRansition  
 STATus:QUEStionable:VOLTagE:CONDition?  
 STATus:QUEStionable:VOLTagE:ENABle  
 STATus:QUEStionable:VOLTagE:NTRansition  
 STATus:QUEStionable:VOLTagE:PTRansition  
 STATus:QUEStionable:VOLTagE[:EVENT]?

## SCPI Compliance Information

SYSTem:BEEPer:STATe  
SYSTem:BEEPer[:IMMEdiate]-?  
SYSTem:COMMunicate:GPIB:ADDRess  
SYSTem:COMMunicate:SERial[:RECeive]:BAUD  
SYSTem:COMMunicate:SERial[:RECeive]:BITS  
SYSTem:COMMunicate:SERial[:RECeive]:PACE  
SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECK  
SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]  
SYSTem:COMMunicate:SERial[:RECeive]:SBITS  
SYSTem:COMMunicate:SERial:TRANsmit:PACE  
SYSTem:DATE  
SYSTem:ERRor?  
SYSTem:KEY  
SYSTem:KLOCK  
SYSTem:PRESet  
SYSTem:SET  
SYSTem:TIME  
SYSTem:VERSion?

TRACe[:DATA]  
TRACe:WATERfall[:DATA]

TRIGger[:IMMEdiate]  
TRIGger:LEVel  
TRIGger:SLOPe  
TRIGger:SOURce



**List of Approved Commands**

HCOPY:DESTination  
HCOPY:DEVice:LANGuage (except PHPGI  
HCOPY:DEVice:SPEed  
HCOPY[:IMMEDIATE]  
HCOPY:ITEM:ALL[:IMMEDIATE]  
HCOPY:ITEM:FFEed:STATe  
HCOPY:ITEM:LABel:STATe  
HCOPY:ITEM:LABel:TEXT  
HCOPY:ITEM:TDSTamp:STATe  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:COLor  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:GRATicule:COLor  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:GRATicule[:IMMEDIATE]  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe[:IMMEDIATE]  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:LTYPE  
HCOPY:PAGE:DIMensions:AUTO

MMEMory:NAME

## List of Instrument Specific Commands

ARM:RPM:INCRement  
ARM:RPM:MODE  
ARM:RPM:THReshold  
ARM:TIMer

CALCulate[1|2|3|4]:CFIT:ABORt  
CALCulate[1|2|3|4]:CFIT:COPIY  
CALCulate[1|2|3|4]:CFIT:DATA  
CALCulate[1|2|3|4]:CFIT:DESTination  
CALCulate[1|2|3|4]:CFIT:FREQuency[:AUTO]  
CALCulate[1|2|3|4]:CFIT:FREQuency:STARt  
CALCulate[1|2|3|4]:CFIT:FREQuency:STOP  
CALCulate[1|2|3|4]:CFIT:FSCale  
CALCulate[1|2|3|4]:CFIT[:IMMediate]  
CALCulate[1|2|3|4]:CFIT:ORDer:AUTO  
CALCulate[1|2|3|4]:CFIT:ORDer:POLes  
CALCulate[1|2|3|4]:CFIT:ORDer:ZERos  
CALCulate[1|2|3|4]:CFIT:TDELay  
CALCulate[1|2|3|4]:CFIT:WEIGHt:AUTO  
CALCulate[1|2|3|4]:CFIT:WEIGHt:REGister  
CALCulate[1|2|3|4]:DATA:HEADer:POINts  
CALCulate[1|2|3|4]:GDAPerture:APERture  
CALCulate[1|2|3|4]:LIMit:BEEP[:STATe]  
CALCulate[1|2|3|4]:LIMit:LOWer:CLEar[:IMMediate]  
CALCulate[1|2|3|4]:LIMit:LOWer:MOVE:Y  
CALCulate[1|2|3|4]:LIMit:LOWer:REPort[:DATA]  
CALCulate[1|2|3|4]:LIMit:LOWer:REPort:YDATA  
CALCulate[1|2|3|4]:LIMit:LOWer:SEGMENT  
CALCulate[1|2|3|4]:LIMit:LOWer:SEGMENT:CLEar  
CALCulate[1|2|3|4]:LIMit:LOWer:TRACe[:IMMediate]  
CALCulate[1|2|3|4]:LIMit:UPPer:CLEar[:IMMediate]  
CALCulate[1|2|3|4]:LIMit:UPPer:MOVE:Y  
CALCulate[1|2|3|4]:LIMit:UPPer:REPort[:DATA]  
CALCulate[1|2|3|4]:LIMit:UPPer:REPort:YDATA  
CALCulate[1|2|3|4]:LIMit:UPPer:SEGMENT  
CALCulate[1|2|3|4]:LIMit:UPPer:SEGMENT:CLEar  
CALCulate[1|2|3|4]:LIMit:UPPer:TRACe[:IMMediate]  
CALCulate[1|2|3|4]:MARKer:BAND:STARt  
CALCulate[1|2|3|4]:MARKer:BAND:STOP  
CALCulate[1|2|3|4]:MARKer:COUPLed[:STATe]  
CALCulate[1|2|3|4]:MARKer:DTABLE:CLEar[:IMMediate]  
CALCulate[1|2|3|4]:MARKer:DTABLE:COPIY[1|2|3|4]  
CALCulate[1|2|3|4]:MARKer:DTABLE[:DATA]  
CALCulate[1|2|3|4]:MARKer:DTABLE:X[:DATA]  
CALCulate[1|2|3|4]:MARKer:DTABLE:X:DELeTe  
CALCulate[1|2|3|4]:MARKer:DTABLE:X:INSert

CALCulate[1|2|3|4]:MARKer:DTABLE:X:LABel  
 CALCulate[1|2|3|4]:MARKer:DTABLE:X:SElect[:POINT]  
 CALCulate[1|2|3|4]:MARKer:FUNCTion  
 CALCulate[1|2|3|4]:MARKer:FUNCTion:RESult  
 CALCulate[1|2|3|4]:MARKer:HARMonic:COUNt  
 CALCulate[1|2|3|4]:MARKer:HARMonic:FUNDamental  
 CALCulate[1|2|3|4]:MARKer:MAXimum[:GLOBal]  
 CALCulate[1|2|3|4]:MARKer:MAXimum[:GLOBal]:TRACk  
 CALCulate[1|2|3|4]:MARKer:MAXimum:LEFT  
 CALCulate[1|2|3|4]:MARKer:MAXimum:RIGHT  
 CALCulate[1|2|3|4]:MARKer:MODE  
 CALCulate[1|2|3|4]:MARKer:POSition  
 CALCulate[1|2|3|4]:MARKer:POSition:POINT  
 CALCulate[1|2|3|4]:MARKer:REFerence:X  
 CALCulate[1|2|3|4]:MARKer:REFerence:Y  
 CALCulate[1|2|3|4]:MARKer:SIDeband:CARRier  
 CALCulate[1|2|3|4]:MARKer:SIDeband:COUNt  
 CALCulate[1|2|3|4]:MARKer:SIDeband:INCRement  
 CALCulate[1|2|3|4]:MARKer[:STATe]  
 CALCulate[1|2|3|4]:MARKer:X[:ABSolute]  
 CALCulate[1|2|3|4]:MARKer:X:RELative  
 CALCulate[1|2|3|4]:MARKer:Y[:ABSolute]  
 CALCulate[1|2|3|4]:MARKer:Y:RELative  
 CALCulate[1|2|3|4]:MATH:CONStant[1|2|3|4|5]  
 CALCulate[1|2|3|4]:MATH:DATA  
 CALCulate[1|2|3|4]:MATH:SElect  
 CALCulate[1|2|3|4]:SYNThesis:COPI  
 CALCulate[1|2|3|4]:SYNThesis:DATA  
 CALCulate[1|2|3|4]:SYNThesis:DESTination  
 CALCulate[1|2|3|4]:SYNThesis:FSCale  
 CALCulate[1|2|3|4]:SYNThesis:GAIN  
 CALCulate[1|2|3|4]:SYNThesis[:IMMediate]  
 CALCulate[1|2|3|4]:SYNThesis:SPACing  
 CALCulate[1|2|3|4]:SYNThesis:TDELay  
 CALCulate[1|2|3|4]:SYNThesis:TTYPe  
 CALCulate[1|2|3|4]:UNIT:AMPLitude  
 CALCulate[1|2|3|4]:UNIT:DBReference  
 CALCulate[1|2|3|4]:UNIT:DBReference:IMPedance  
 CALCulate[1|2|3|4]:UNIT:DBReference:USER:LABel  
 CALCulate[1|2|3|4]:UNIT:DBReference:USER:REFerence  
 CALCulate[1|2|3|4]:UNIT:MECHanical  
 CALCulate[1|2|3|4]:UNIT:VOLTag  
 CALCulate[1|2|3|4]:UNIT:X  
 CALCulate[1|2|3|4]:UNIT:X:ORDer:FACTor  
 CALCulate[1|2|3|4]:UNIT:X:USER:FREQuency:FACTor  
 CALCulate[1|2|3|4]:UNIT:X:USER:FREQuency:LABel  
 CALCulate[1|2|3|4]:UNIT:X:USER:TIME:FACTor  
 CALCulate[1|2|3|4]:UNIT:X:USER:TIME:LABel  
 CALCulate[1|2|3|4]:WATERfall:COUNt  
 CALCulate[1|2|3|4]:WATERfall[:DATA]

## SCPI Compliance Information

CALCulate[1|2|3|4]:WATERfall:SLICe:COPIY  
CALCulate[1|2|3|4]:WATERfall:SLICe:SELEct  
CALCulate[1|2|3|4]:WATERfall:SLICe:SELEct:POINt  
CALCulate[1|2|3|4]:WATERfall:TRACe:COPIY  
ALCulate[1|2|3|4]:WATERfall:TRACe:SELEct  
CALCulate[1|2|3|4]:WATERfall:TRACe:SELEct:POINt

DISPlay:BODE  
DISPlay:BRIGHtness  
DISPlay:ERRor  
DISPlay:EXTernal[:STATe]  
DISPlay:GPIB:ECHO  
DISPlay:PROGram:KEY:BOX  
DISPlay:PROGram:KEY:BRACket  
DISPlay:PROGram:VECTor:BUFFer[:STATe]  
DISPlay:RPM[:STATe]  
DISPlay:STATe  
DISPlay:TCAPture:ENVelope[:STATe]  
DISPlay:VIEW  
DISPlay[:WINDow[1|2|3|4]]:DTABLE:MARKer[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:DTABLE[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:LIMit:STATe  
DISPlay[:WINDow[1|2|3|4]]:POLar:CLOCKwise  
DISPlay[:WINDow[1|2|3|4]]:POLar:ROTation  
DISPlay[:WINDow[1|2|3|4]]:TRACe:APOWer[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:TRACe:BPOWer[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:TRACe:LAbel  
DISPlay[:WINDow[1|2|3|4]]:TRACe:LAbel:DEFault[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:TRACe:X:MATCh[1|2|3|4]  
DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALE]:AUTO  
DISPlay[:WINDow[1|2|3|4]]:TRACe:Y:MATCh[1|2|3|4]  
DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALE]:CENTer  
DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALE]:REFerence  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:BASeline  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:BOTTom  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:COUNt  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:HEIGHt  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:HIDDen  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:SKEW  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:SKEW:ANGLE  
DISPlay[:WINDow[1|2|3|4]]:WATERfall[:STATe]  
DISPlay[:WINDow[1|2|3|4]]:WATERfall:TOP

HCOPY::COLor:DEFault  
HCOPY:ITEM:LAbel:COLor 0~16  
HCOPY:ITEM:TDSTamp:FORMat  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:LIMit:LTYPe  
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:MARKer:COLor

HCOpy:ITEM[:WINDow[1|2|3|4]]:TRACe:MARKer[:IMMediate]  
 HCOpy:ITEM[:WINDow[1|2|3|4]]:TRACe:MARKer:REFeRence[:IMMediate]  
 HCOpy:PAGE:DIMensions:USER:LLEFt  
 HCOpy:PAGE:DIMensions:USER:URIGHt  
 HCOpy:PLOT:ADDReSS  
 HCOpy:PRINt:ADDReSS  
 HCOpy:TITLe[1|2]

INPut[1|2|3|4]:REFeRence:DIRectiOn 0~32767  
 INPut[1|2|3|4]:REFeRence:POINt 0~32767

MEMory:CATalog:NAME?

MMEMory:DISK:ADDReSS  
 MMEMory:DISK:UNIT  
 MMEMory:FSYStem?  
 MMEMory:LOAD:CFIT  
 MMEMory:LOAD:CONTinue  
 MMEMory:LOAD:CONTinue?  
 MMEMory:LOAD:DTABLE:TRACe[1|2|3|4]  
 MMEMory:LOAD:LIMit:LOWer:TRACe[1|2|3|4]  
 MMEMory:LOAD:LIMit:UPPer:TRACe[1|2|3|4]  
 MMEMory:LOAD:MATH  
 MMEMory:LOAD:PROGram  
 MMEMory:LOAD:SYNThesis  
 MMEMory:LOAD:TCAPture  
 MMEMory:LOAD:WATERfall  
 MMEMory:MDIRectory  
 MMEMory:STORE:CFIT  
 MMEMory:STORE:CONTinue  
 MMEMory:STORE:CONTinue?  
 MMEMory:STORE:DTABLE:TRACe[1|2|3|4]  
 MMEMory:STORE:LIMit:LOWer:TRACe[1|2|3|4]  
 MMEMory:STORE:LIMit:UPPer:TRACe[1|2|3|4]  
 MMEMory:STORE:MATH  
 MMEMory:STORE:PROGram  
 MMEMory:STORE:PROGram:FORMat  
 MMEMory:STORE:SYNThesis  
 MMEMory:STORE:TCAPture  
 MMEMory:STORE:WATERfall

OUTPut:FILTer[:LPASs][:STATe]

PROGram:EDIT:ENABle  
 PROGram:EXPLicit:LABEL

## SCPI Compliance Information

PROGram[:SElected]:LABel

[SENSe:]AVERAge:CONFidence  
[SENSe:]AVERAge:HOLD  
[SENSe:]AVERAge:IMPulse  
[SENSe:]AVERAge:IRESult:RATE  
[SENSe:]AVERAge:IRESult[:STATe]  
[SENSe:]AVERAge:PREView  
[SENSe:]AVERAge:PREView:ACcept  
[SENSe:]AVERAge:PREView:REJect  
[SENSe:]AVERAge:PREView:TIME  
[SENSe:]AVERAge:TIME  
[SENSe:]AVERAge:TYPE  
[SENSe:]FEED  
[SENSe:]FREQuency:BLOCKsize  
[SENSe:]FREQuency:RESolution:AUTO:MCHange  
[SENSe:]FREQuency:RESolution:AUTO:MINimum  
[SENSe:]FREQuency:RESolution:OCTave  
[SENSe:]HISTogram:BINS  
[SENSe:]ORDer:MAXimum  
[SENSe:]ORDer:RESolution  
[SENSe:]ORDer:RESolution:TRACk  
[SENSe:]ORDer:RPM:MAXimum  
[SENSe:]ORDer:RPM:MINimum  
[SENSe:]ORDer:TRACk[1|2|3|4|5]  
[SENSe:]ORDer:TRACk[1|2|3|4|5]:STATe  
[SENSe:]REFerence  
[SENSe:]REJect:STATe  
[SENSe:]SWEep:OVERlap  
[SENSe:]SWEep:STIME  
[SENSe:]TCAPture:ABORt  
[SENSe:]TCAPture:DELeTe  
[SENSe:]TCAPture[:IMMediate]  
[SENSe:]TCAPture:LENGth  
[SENSe:]TCAPture:MALLocate  
[SENSe:]TCAPture:STARt[1|2|3|4]  
[SENSe:]TCAPture:STOP[1|2|3|4]  
[SENSe:]TCAPture:TACHometer:RPM:MAXimum  
[SENSe:]TCAPture:TACHometer[:STATe]  
[SENSe:]VOLTagE[1|2|3|4][:DC]:RANGe:AUTO:DIRection  
[SENSe:]VOLTagE[1|2|3|4][:DC]:RANGe:UNIT:USER:LABel  
[SENSe:]VOLTagE[1|2|3|4][:DC]:RANGe:UNIT:USER:SFACTOR  
[SENSe:]VOLTagE[1|2|3|4][:DC]:RANGe:UNIT:USER[:STATe]  
[SENSe:]VOLTagE[1|2|3|4][:DC]:RANGe:UNIT:XDCR:LABel  
[SENSe:]WINDow[1|2|3|4]:ORDer:DC

SOURce:BURSt  
 SOURce:USER:CAPTure  
 SOURce:USER[:REGister]  
 SOURce:USER:REPeat  
 SOURce:VOLTagE[:LEVel]:AUTO  
 SOURce:VOLTagE[:LEVel][:IMMediate]:OFFSet  
 SOURce:VOLTagE[:LEVel]:REFerence  
 SOURce:VOLTagE[:LEVel]:REFerence:CHANnel  
 SOURce:VOLTagE[:LEVel]:REFerence:TOLerance  
 SOURce:VOLTagE:LIMit:INPut  
 STATus:QUEStionable:LIMit:CONDition  
 STATus:QUEStionable:LIMit:ENABle  
 STATus:QUEStionable:LIMit[:EVENT]  
 STATus:QUEStionable:LIMit:NTRansition  
 STATus:QUEStionable:LIMit:PTRansition  
 STATus:USER:ENABle  
 STATus:USER[:EVENT]  
 STATus:USER:PULSe

SYSTem:FAN[:STATe]  
 SYSTem:POWer:SOURce  
 STATus:DEVIce:CONDition?  
 STATus:DEVIce:ENABle  
 STATus:DEVIce[:EVENT]  
 STATus:DEVIce:NTRansition  
 STATus:DEVIce:PTRansition

TEST:LOG:CLEar  
 TEST:LONG  
 TEST:LONG:RESult

TRACe:X[:DATA]?  
 TRACe:X:UNIT?  
 TRACe:Z[:DATA]?  
 TRACe:Z:UNIT?

TRIGger:EXTernal:FILTer[:LPAS][:STATe]  
 TRIGger:EXTernal:LEVel  
 TRIGger:EXTernal:RANGe  
 TRIGger:LEVel:TTL  
 TRIGger:STARt[1|2|3|4]  
 TRIGger:TACHometer:HOLDoff  
 TRIGger:TACHometer:LEVel  
 TRIGger:TACHometer:PCOunt  
 TRIGger:TACHometer:RANGe

## SCPI Compliance Information

TRIGger:TACHometer[:RPM]  
TRIGger:TACHometer:SLOPe



## Introduction to the Command Reference

---



## Introduction to the Command Reference

---

The Command Reference chapters describe all of the Agilent 35670A's GPIB commands. Each command has the following:

A brief description of the command. This one- or two-line description appears just below the heading.

A syntax description. This consists of two fields. One field specifies whether the command has only a command form, only a query form, or both. The other field shows you the syntax expected by the analyzer's GPIB command parser. At the end of this chapter is a detailed description for the elements appearing in the syntax description. Additional information about message syntax is also available in the GPIB Programmer's Guide.

<b>FORMat[:DATA]</b>	<b>command/query</b>
Specifies the data type and date encoding to be used during transfers of a data block.	1
<b>Command Syntax:</b> FORMat[:DATA] ASCii REAL, [<number> <bound>]	
<number> ::= a real number (NRf data)	2
limits: 3:64	
<bound> ::= MAX MIN	
<b>Example Statements:</b>	
OUTPUT 711;"FORMAT:DATA ASCii, 8"	3
OUTPUT 711;"FORM REAL, 64"	
<b>Query Syntax:</b>	FORMat[:DATA]? 4
<b>Return Format:</b>	ASC,NR1 REAL,NR1 4
<b>Attribute Summary:</b>	Option: not applicable Synchronization Required: no Preset State: ASC 12 SCPI Compliance: confirmed 5
<b>Description:</b>	
FORM:DATA only affects data transfers initiated by the following commands:	
■ CALC:DATA?	6
■ CALC:CFIT:DATA?	
■ CALC:LIM:LOW:REP?	
■ CALC:LIM:LOW:SEGM	

Example statements. This field appears at the end of the syntax description. It contains two BASIC output statements that use the command.

A return format description. This field is only used if the command has a query form. It tells you how data is returned in response to the query.

An attribute summary. This field indicates whether one of the analyzer's options must be installed; identifies overlapped commands requiring synchronization; defines the command's preset state and specifies compliance with SCPI. A "confirmed" command complies with SCPI 1992. An "approved" command complies with SCPI 1993. An "instrument-specific" command does not conform to the SCPI standard.

A detailed description. This field contains additional information about the command.

---

## Finding the Right Command

- If you can not find a command you have seen in a program, remember that commands can omit implied mnemonics.

For example, the command `SENSE:FREQUENCY:CENTer 10000 HZ` contains the implied mnemonic `SENSe`. `SENSe` can be omitted to create the equivalent command `FREQUENCY:CENTer 10000 HZ`. (See “Implied Mnemonics” in the GPIB Programmer’s Guide.) You will not find an entry for `FREQUENCY:CENTer`—or any other command that omits an implied mnemonic—in the Command Reference. You will find the `FREQUENCY:CENTer` command in the `SENSe` chapter.

- If you do not find a command where you expect it, try scanning the command list in appendix A for the equivalent command that contains the implied mnemonic.

Each command has a brief description. After you locate the equivalent command, you can find a more detailed description in the corresponding command reference chapter.

- If you are looking for a command that accesses a particular function, use the index.

For example, if you want to find the command that changes the analyzer’s center frequency, look for “center frequency” in the index. It sends you to the page that describes the `SENSe:FREQUENCY:CENTer` command.

- If you are familiar with front panel operation of the analyzer, use Appendix B.

It provides a cross reference of the analyzer’s hardkeys and softkeys and their equivalent GPIB commands.

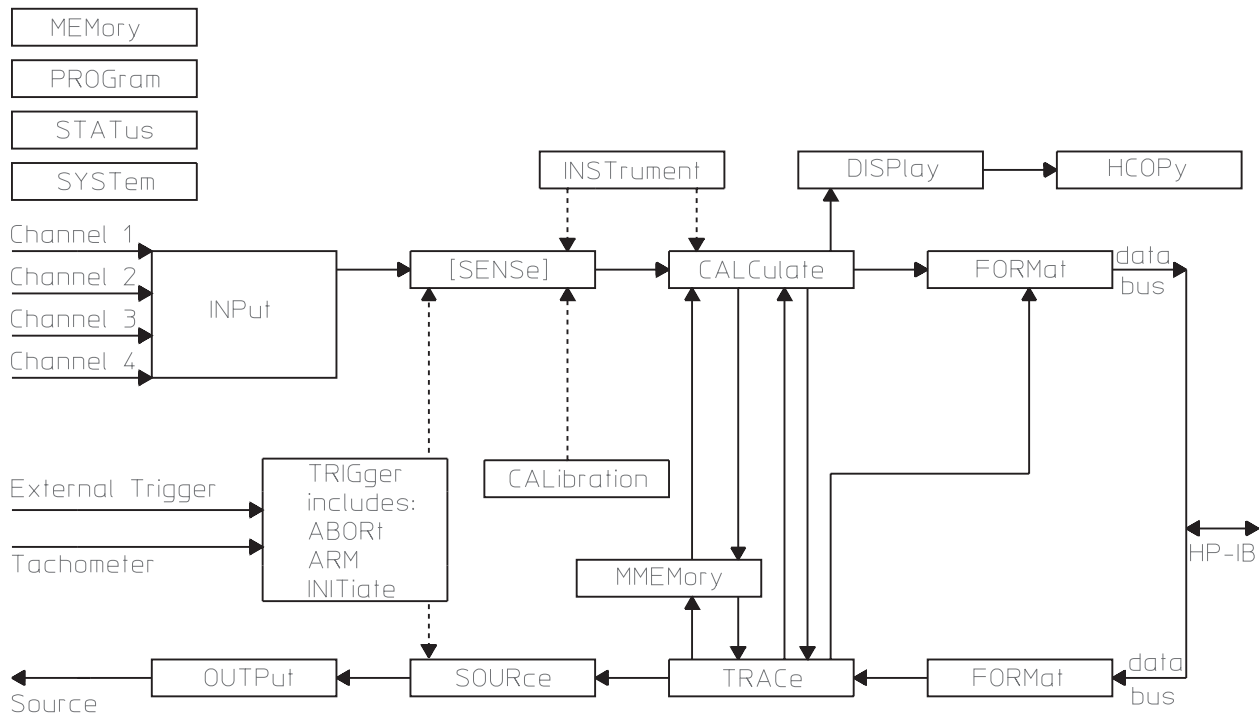
- If you have an analyzer, use the GPIB Echo facility.

GPIB Echo displays the corresponding GPIB command for front panel operations. It displays the most abbreviated form of the command. To turn on the facility, press

[ Local/GPIB ] [ GPIB ECHO ON OFF ] to highlight **ON**.

## Finding the Right Command

Figure 2-2 shows the flow of control and measurement data through the Agilent 35670A. The SCPI subsystems appear in the block diagram. Use figure 2-2 to help identify the subsystem containing a command. For example, the location of the SENSE subsystem indicates it contains commands which determine how measurement data is acquired. The use of the brackets ([ ]) indicates SENSE is an implied mnemonic.



2-2. Flow of Control and Measurement Data in the Agilent 35670A

---

## Command Syntax in the Agilent 35670A

The Agilent 35670A uses program messages and response messages to communicate with other devices on the GPIB. This section describes the syntax elements used in the command reference chapters. It also describes the general syntax rules for both kinds of messages.

---

**Note** For a more detailed discussion of message syntax, including example program listings, refer to the GPIB Programmer's Guide.

---

### Special Syntactic Elements

Several syntactic elements have special meanings:

- colon (:) — When a command or query contains a series of keywords, the keywords are separated by colons. A colon immediately following a keyword tells the command parser that the program message is proceeding to the next level of the command tree. A colon immediately following a semicolon tells the command parser that the program message is returning to the base of the command tree. For more information, see “Programming with GPIB Commands” in the GPIB Programmer's Guide.
- semicolon (;) — When a program message contains more than one command or query, a semicolon is used to separate them from each other. For example, if you want to autorange the analyzer's inputs and then start a measurement using one program message, the message would be:
 

```
SENSE:VOLT:RANGE:AUTO ONCE;:ABORT;:INITIATE:IMMEDIATE
```
- comma (,) — A comma separates the data sent with a command or returned with a response. For example, the SYSTEM:TIME command requires three values to set the analyzer's clock: one for hours, one for minutes, and one for seconds. A message to set the clock to 8:45 AM would be:
 

```
SYSTEM:TIME 8,45,0
```
- <WSP> — One white space is required to separate a program message (the command or query) from its parameters. For example, the command “SYSTEM:TIME 8,45,0” contains a space between the program header (SYSTEM:TIME) and its program data (8,45,0). White space characters are not allowed within a program header.

For more information, see “GPIB Message Syntax” in the GPIB Programmer's Guide.

## Conventions

Syntax and return format descriptions use the following conventions:

- `< >` Angle brackets enclose the names of items that need further definition. The definition will be included in accompanying text. In addition, detailed descriptions of these elements appear at the end of this chapter.
- `::=` “is defined as” When two items are separated by this symbol, the second item replaces the first in any statement that contains the first item. For example, `A::=B` indicates that B replace A in any statement that contains A.
- `|` “or” When items in a list are separated by this symbol, one and only one of the items can be chosen from the list. For example, `A|B` indicates that A or B can be chosen, but not both.
- `...` An ellipsis (trailing dots) is used to indicate that the preceding element may be repeated one or more times.
- `[ ]` Square brackets indicate that the enclosed items are optional.
- `{ }` Braces are used to group items into a single syntactic element. They are most often used to enclose lists and to enclose elements that are followed by an ellipsis.

Although the analyzer is not case sensitive, the case of letters in the command keyword is significant in the Command Reference. Keywords that are longer than four characters can have a short form or a long form. The analyzer accepts either form. Upper-case letters show the short form of a command keyword. For more information, see “Programming with GPIB Commands” in the GPIB Programmer’s Guide .

The analyzer is sensitive to white space characters. White space characters are not allowed within command keywords. They are only allowed when they are used to separate a command and a parameter.

A message terminator is required at the end of a program message or a response message. Use `<NL>`, `<^END>`, or `<NL> <^END>` as the program message terminator. The word `<NL>` is an ASCII new line (line feed) character. The word `<^END>` means that End or Identify (EOI) is asserted on the GPIB interface at the same time the preceding data byte is sent. Most programming languages send these terminators automatically. For example, if you use the BASIC OUTPUT statement, `<NL>` is automatically sent after your last data byte. If you are using a PC, you can usually configure your system to send whatever terminator you specify.

For more information about terminating messages, see “Programming with GPIB Commands” in the GPIB Programmer’s Guide .



## Syntax Descriptions

Syntax descriptions in the command reference chapters use the following elements:

### <BLOCK>

This item designates block data. There are three kinds of block data; binary-definite-length-block data, binary-indefinite-length-block data, and ASCII data. The analyzer always returns binary definite-length-block data or ASCII data in response to queries.

When you send block data to the analyzer, you can send it either as definite-length block or indefinite-length block data or as ASCII data.

If you are sending binary data using indefinite-length syntax, <BLOCK> takes the following form:

```
<BLOCK> ::= #0<data_byte>[,<data_byte>] . . .<NL><^END>
```

```
<data_byte> ::= unsigned 8-bit data
```

```
<NL> ::= new line (line feed) character, ASCII decimal 10
```

```
<^END> ::= GPIB END message (EOI set true)
```

If you are sending binary data using definite-length syntax, <BLOCK> takes the following form:

```
<BLOCK> ::= #<byte><length_bytes><data_byte>[<data_byte>] . . .
```

```
<byte> ::= number of length bytes to follow (ASCII encoded)
```

```
<length_bytes> ::= number of data bytes to follow (ASCII encoded)
```

```
<data_byte> ::= unsigned 8-bit data
```

If you are sending ASCII data, <BLOCK> takes the following form:

```
<BLOCK> ::= integer[,integer]. . .<NL>
```

or

```
<BLOCK> ::= floating-point-numeric[,floating-point-numeric]. . .<NL>
```

See "GPIB Message Syntax" in the *GPIB Programmer's Guide* for more information about block data.

### <CHAR>

This item designates a string of ASCII characters. There are no delimiters. Usually, the string is from an explicit set of responses. Maximum length is 12 characters.

### <CMDSTR>

This string specifies the type of measurement data. Channels 3 and 4 are only available with Option AY6.

D[1|2] . . . [8] selects a data register

TCAPture [1|2|3|4] selects a time capture buffer

W[1|2]. . . [8] selects a waterfall register

XFRrequency:POWer [1|2|3|4] selects power spectrum

XFRrequency:POWer:COHerence [1,2|1,3|1,4|3,4] selects coherence

XFRrequency:POWer:CROSSs [1,2|1,3|1,4|3,4] selects cross spectrum

XFRrequency:POWer:LINear [1|2|3|4] selects linear spectrum  
XFRrequency:POWer:RATio [2,1|3,1|4,1|4,3] selects frequency response  
XFRrequency:POWer:VARiance [1|2|3|4] selects normalized variance  
XORDer:TRACk [1|2|3|4],[1,1|2|3|4|5] selects order track  
XRPM:PROFile selects RPM profile  
XTIME:CORRelation [1|2|3|4] selects auto correlation  
XTIME:CORRelation:CROSSs [1,2|1,3|1,4|3,4] selects cross correlation  
XTIME:VOLTag:e:CDF [1|2|3|4] selects cumulative density function  
XTIME:VOLTag:e:HISTogram [1|2|3|4] selects histogram  
XTIME:VOLTag:e:PDF [1|2|3|4] selects probability density function  
XTIME:VOLTag:e:WINDow [1|2|3|4] selects windowed time  
XVOLtag:e:VOLTag:e [1,2|1,3|1,4|3,4] selects orbit

<DEF\_BLOCK>

This item designates definite-length-block data which takes the following form if the data is binary encoded (FORMat:DATA REAL command):

<DEF\_BLOCK> ::= #<byte><length\_bytes><data\_byte>[<data\_byte>] . . .

<byte> ::= number of length bytes to follow (ASCII encoded)

<length\_bytes> ::= number of data bytes to follow (ASCII encoded)

<data\_byte> ::= unsigned 8-bit data

If the data is ASCII encoded (FORMat:DATA ASCii command):

<DEF\_BLOCK> ::= integer[,integer] . . . <NL>

or

<DEF\_BLOCK> ::= floating-point-numeric[,floating-point-numeric] . . . <NL>

See "GPIB Message Syntax" in the *GPIB Programmer's Guide* for more information about block data.

<EXPR>

This item designates an expression for a math function.

<EXPR> ::= ([<expr\_element>]...)

<expr\_element> ::= See the operations and operands listed below:

**Operations:**

AWEIGHT Apply A-weight filter

BWEIGHT Apply B-weight filter

CWEIGHT Apply C-weight filter

CONJ Complex Conjugate

DIFF Differentiate

DJOM Divide by  $j \theta$   
 EXP Exponential  
 FFT Fast Fourier Transform  
 IFFT Inverse Fast Fourier Transform  
 INTEG Integrate  
 IMAG Imaginary Part  
 LN Natural Logarithm  
 MAG Magnitude  
 PSD Power Spectral Density  
 REAL Real Part  
 SQRT Square Root  
 XJOM multiply by  $j \theta$

+ Add

- Subtract

\* Multiply

/ Divide

**Operands :**

D1|D2|. . . D8 Contents of data registers

F1|F2|. . . F5 Contents of function registers

K1|K2|. . . K5 Contents of constant registers

Measurement Data (depends on instrument mode)

[1|2|3|4] specifies which channel the data was taken from. Channels 3 and 4 are only available with Option AY6.

PSPEC[1|2|3|4] Power Spectrum

LSPEC[1|2|3|4] Linear Spectrum

TIME[1|2|3|4] Time Data

WTIME[1|2|3|4] Windowed Time Data

FRES[21|31|41|43] Frequency Response

CSPEC[21|31|41|43] Cross Spectrum

COH[21|31|41|43] Coherence

HIST[1|2|3|4] Histogram (INST:SEL HIST only)

PDF[1|2|3|4] Probability Density Function (INST:SEL HIST only)

CDF[1|2|3|4] Cumulative Density Function (INST:SEL HIST only)

TIME[1|2|3|4] Unfiltered Time (INST:SEL HIST only)

## Command Syntax in the Agilent 35670A

ACORR[1|2|3|4] Autocorrelation (INST:SEL CORR only)  
XCORR[21|31|41|43] Cross Correlation (INST:SEL CORR only)  
CPOW[1|2|3|4] Composite Power (INST:SEL ORD only; Option 1D0)  
TRACK[1|2|3|4|5] (INST:SEL ORD only; Option 1D0)  
    [1|2|3|4|5] specifies order track  
TIME[1|2|3|4] Resampled Time (INST:SEL ORD only; Option 1D0)  
RPM RPM Profile (INST:SEL ORD only; Option 1D0)  
NVAR[1|2|3|4] Normalized Variance (INST:SEL SINE only; Option 1D2)

<FILE>

This string is used to describe the name of a file. It does not include any disk drive information.

The valid character set for <FILE> depends on the disk format.

DOS file names are limited to 8 ASCII characters followed by a period and a 3 ASCII character extension. The period and extension are not required. File names are not case sensitive.

LIF file names are limited to 10 ASCII characters which may include all characters except “:” “<” and “|”. The first character must be a letter. File names *are* case sensitive.

<FILENAME>

This string is used to describe the name of a file on the default disk, on the specified disk, or the specified disk and DOS directory. The allowed form is:

“[PATH]filename”

where PATH must be replaced with:

“[MSIS:][\DOS\_DIR\]”

where MSIS: must be replaced with:

RAM: which selects the volatile RAM disk.

NVRAM: which selects the non-volatile RAM disk.

INT: which selects the internal disk drive.

EXT[,<select\_code>[,<unit\_number>]]: which selects an external disk drive.

The brackets are not literal in the parameter. They indicate that the disk drive designation is optional. If the disk drive is not specified, the default disk drive is used. Select the default specifier with the MMEM:MSIS command.

DOS\_DIR specifies a directory name (DOS only). DOS directory names are limited to ASCII characters. The forward slash (/) may be used as a directory separator instead of the backward slash (\).

The brackets are not literal in the parameter. They indicate that the directory designation is optional. If the director is not specified, the default directory is used. Select the default specifier with the MMEM:MSIS command.

The valid character set for <FILENAME> depends on the disk format.

DOS file names are limited to 8 ASCII characters followed by a period and a 3 ASCII character extension. The period and extension are not required. File names are *not* case sensitive.

LIF file names are limited to 10 ASCII characters which may include all characters except “:” “<” and “|”. The first character must be a letter. File names *are case sensitive*.

#### <MSINAME>

This item specifies the mass storage device. It is a <STRING> with one of the following values:

“RAM:”

“NVRAM:”

“INT:”

“EXT[,<select\_code>[,<unit\_number>]]:”

#### <MEMNAME>

This item specifies a single file or an entire mass storage device. It takes the same form as <FILENAME> except either the name of the file, the name of the mass storage device or both must be present. It can take one of the following forms:

“[MSIS:][DOS\_DIR:]filename”

“MSIS:”

“MSIS:[DOS\_DIR:]filename”

See the description for <FILENAME> for the valid mass storage device specifiers and the valid file name character set. The optional directory specifier (DOS only) may be included with the filename.

#### <PROGRAM>

This item designates an Instrument BASIC program.

Load and save Instrument BASIC using the indefinite-length-block data format. The simplest way to load an Instrument BASIC program into the analyzer is to send the command (PROG[:SElected]:DEFine) followed by #0, followed by all the characters making up the program, including line numbers and linefeeds at the end of each program statement.

Terminate the entire command with new line (line feed) character (ASCII decimal 10) and <^END> (the GPIB END message (EOI set true) while the last byte of data is on the bus).

#### <STRING>

This item specifies any 8-bit characters delimited by single quotes or double quotes. The beginning and ending delimiter must be the same. If the delimiter character is in the string, it must be entered twice. (For example, to get “EXAMPLE”, enter “”EXAMPLE””).



## Common Commands

---





## Common Commands

---

This chapter describes all of the IEEE 488.2 common commands implemented by the Agilent 35670A. An important property of all common commands is that you can send them without regard to a program message's position in the GPIB command tree.

For more information on the GPIB command tree, see chapter 2, "Programming with GPIB Commands," in the GPIB Programmer's Guide .

**\*CAL?**

**query**

Calibrates the analyzer and returns the result.

Query Syntax:           \*CAL?

Example Statements:     OUTPUT 711;"\*CAL?"  
                          OUTPUT 711;"\*cal?"

Return Format:           Integer

Attribute Summary:     Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

The analyzer performs a full calibration when you send this query. If the calibration completes without error, the analyzer returns 0. If the calibration fails, the analyzer returns 1.

This query is the same as the CAL:ALL? query.

---

**\*CLS****command**

Clears the Status Byte by emptying the error queue and clearing all event registers.

Command Syntax:           \*CLS

Example Statements:        OUTPUT 711;"\*Cls"  
                              OUTPUT 711;"\*CLS"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This command clears the Status Byte register. It does so by emptying the error queue and clearing (setting to 0) all bits in the event registers of the following register sets:

- User Status
- Device State
- Questionable Voltage
- Limit Fail
- Questionable Status
- Standard Event
- Operation Status

In addition, \*CLS cancels any preceding \*OPC command or query. This ensures that bit 0 of the Standard Event register will not be set to 1 and that a response will not be placed in the analyzer's output queue when pending overlapped commands are completed.

\*CLS does not change the current state of enable registers or transition filters.

---

**Note**                    To guarantee that the Status Byte's Message Available and Master Summary Status bits are cleared, send \*CLS immediately following a Program Message Terminator.

---

See chapter 1 for more information on the Status Byte register.

**\*ESE****command/query**

Sets bits in the Standard Event enable register.

Command Syntax:            \*ESE <number>|<bound>

                              <number> ::= a real number (NRf data)  
  limits: 0:255

                              <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"\*ese 128"  
                              OUTPUT 711;"\*ESE 60"

Query Syntax:              \*ESE?

Return Format:              Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: dependent on setting of \*PSC  
                              SCPI Compliance: confirmed

Description:

This command allows you to set bits in the Standard Event enable register. Assign a decimal weight to each bit you want set (to 1) according to the following formula:

$$2^{(\text{bit\_number})}$$

with acceptable values for bit\_number being 0 through 7. Add the weights and then send the sum with this command.

When an enable register bit is set to 1, the corresponding bit of the Standard Event event register is enabled. All enabled bits are logically ORed to create the Standard Event summary, which reports to bit 5 of the Status Byte. Bit 5 is only set to 1 if both of the following are true:

- One or more bits in the Standard Event event register are set to 1.
- At least one set bit is enabled by a corresponding bit in the Standard Event enable register.

The setting last specified with \*ESE is saved in nonvolatile memory. It can be recalled at power-up, depending on the setting of the Power-on Status Clear flag (set with \*PSC). When the flag is 0 at power-up, all bits in the Standard Event enable register are set according to the saved \*ESE value. When the flag is 1 at power-up, all bits in the Standard Event enable register are initialized to 0.

The query returns the current state of the Standard Event enable register. The state is returned as a sum of the decimal weights of all set bits.

For more information on the Standard Event register set, see chapter 1.

**\*ESR?****query**

Reads and clears the Standard Event event register.

Query Syntax:           \*ESR?

Example Statements:     OUTPUT 711;"\*esr?"  
                          OUTPUT 711;"\*Esr?"

Return Format:           Integer

Attribute Summary:     Option:   not applicable  
                          synchronization Required:  no  
                          Preset State:   +0  
                          SCPI Compliance:  confirmed

Description:

This query returns the current state of the Standard Event event register. The state is returned as a sum of the decimal weights of all set bits. The decimal weight for each bit is assigned according to the following formula:

$$2^{(\text{bit\_number})}$$

with acceptable values for bit\_number being 0 through 7.

The query clears the register after it reads the register.

A bit in this register is set to 1 when the condition it monitors becomes true. A set bit remains set, regardless of further changes in the condition it monitors, until one of the following occurs:

- You read the register with this query.
- You clear all event registers with the \*CLS command.

For more information on the Standard Event register set, see chapter 1.

**\*IDN?**

**query**

Returns a string that uniquely identifies the analyzer.

Query Syntax:               \*IDN?

Example Statements:        OUTPUT 711;"\*IDN?"  
                              OUTPUT 711;"\*idn?"

Return Format:              Agilent Technologies,35670A,<serial\_number><software\_re-  
                              vision>

Attribute Summary:         Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:   instrument dependent  
                              SCPI Compliance:  confirmed

Description:

The response to this query uniquely identifies your analyzer.

**\*OPC****command/query**

Sets or queries completion of all pending overlapped commands.

Command Syntax:           \*OPC

Example Statements:        OUTPUT 711;"\*OpC"  
                              OUTPUT 711;"\*OPC"

Query Syntax:             \*OPC?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

Some commands are processed sequentially by the analyzer. A sequential command holds off the processing of subsequent commands until it has been completely processed. However, some commands do not hold off the processing of subsequent commands. These commands are called overlapped commands. At times, overlapped commands require synchronization. The Attribute Summary for each command indicates whether it requires synchronization.

The analyzer uses the No Pending Operation (NPO) flag to keep track of overlapped commands that are still pending (that is, not completed). The NPO flag is reset to 0 when an overlapped command is pending. It is set to 1 when no overlapped commands are pending. You cannot read the NPO flag directly, but you can use \*OPC and \*OPC? to tell when the flag is set to 1.

If you use \*OPC, bit 0 of the Event Status event register is set to 1 when the NPO flag is set to 1. This allows the analyzer to generate a service request when all pending overlapped commands are completed (assuming you have enabled bit 0 of the Event Status register and bit 5 of the Status Byte register).

If you use \*OPC?, +1 is placed in the output queue when the NPO flag is set to 1. This allows you to effectively pause the controller until all pending overlapped commands are completed. It must wait until the response is placed in the queue before it can continue.

**Note**            The \*CLS and \*RST commands cancel any preceding \*OPC command or query. Pending overlapped commands are still completed, but you can no longer determine when.  
Two GPIB bus management commands—Device Clear (DCL) and Selected Device Clear (SDC)—also cancel any preceding \*OPC command or query.

---

A list of overlapped commands requiring synchronization appear in chapter 1. See “Synchronization” in the GPIB Programmer’s Guide for additional information about the \*OPC command and the \*OPC? query.



**\*OPT?****query**

Returns a string that identifies the analyzer's option configuration.

Query Syntax:               \*OPT?

Example Statements:        OUTPUT 711;"\*opt?"  
                              OUTPUT 711;"\*Opt?"

Return Format:               "1D1,1D2,1D3,1D4,1C2,AN2,AY6"

Attribute Summary:         Option:   not applicable  
                              Synchronization Required: no  
                              Preset State:  instrument dependent  
                              SCPI Compliance: confirmed

Description:

The response to this query identifies the analyzer's option configuration. For example, if your analyzer has Computed Order Tracking installed, it returns 1D0 to this query. Options are identified by the following:

- 1D0 Computed Order Tracking
- 1D1 Real Time Octave Measurements
- 1D2 Swept Sine Measurements
- 1D3 Curve Fit / Synthesis
- 1D4 Arbitrary Source
- 1C2 Instrument BASIC
- AY6 Add 2 Input Channels
- UFF Add 1 MByte Nonvolatile RAM
- AN2 Add 4 MBytes Memory
- UFC Add 8 MBytes Memory

The query returns a null string ("") if special options are not installed in the analyzer.

**\*PCB**

**command**

Sets the pass-control-back address.

Command Syntax:           \*PCB <primary\_address>, [<secondary\_address>]

    <primary\_address> ::= <number>|<step>|<bound>

    <secondary\_address> ::= <number>|<step>|<bound>

Example Statements:        OUTPUT 711;"\*PCB 11"  
                              OUTPUT 711;"\*PCB 19"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

Description:

Use this command to specify the address of your controller before you pass control of the GPIB to the analyzer. When the analyzer completes the operation that required it to have control of the bus, it automatically passes control back to the controller at the specified address.

The optional second number is only used for controllers that support extended addressing. It is ignored by the analyzer.

The address last specified with this command is saved in nonvolatile memory, so it is not affected when you turn the analyzer off and on. It is also not affected by the \*RST command.



**\*RST**

**command**

Executes a device reset.

Command Syntax:            \*RST

Example Statements:        OUTPUT 711;"\*RST"  
                              OUTPUT 711;"\*rst"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   confirmed

Description:

This command returns the analyzer to a reset state. In addition, \*RST cancels any pending \*OPC command or query.

The reset state is similar to the preset state. The preset state of each command is listed in the Attribute Summary. In some cases, however, a command's reset state differs from its preset state. These commands (and their reset states) are listed below.

- CALibration:AUTO is set to 0 (OFF).
- FREQuency:SPAN:LINK is set to CENTER.
- FREQuency:RESolution:AUTO is ON.
- CALCulate:GDAPerture:APERture is set to 0.25 PCT.
- HCOpy:PAGE:DIMensions:USER:LLEft is set to +332, +1195.
- HCOpy:PAGE:DIMensions:USER:URIGHt is set to +9155, +7377.

---

**Note**                    This command is not equivalent to a front panel preset. Send SYST:PRES if you want to send a command that is equivalent to a front panel preset.

---

The following are not affected by this command:

- The state of the Power-on Status Clear flag.
- The state of all enable and transition registers.
- The GPIB input and output queues.
- The time and date (SYST:TIME and SYST:DATE).
- The GPIB address settings (SYST:COMM:GPIB:ADDR, HCOP:PLOT:ADDR, HCOP:PRIN:ADDR).
- The GPIB controller capability setting.
- The default disk selection (MMEM:MSIS).
- The serial interface (RS-232-C) parameters (set with SYST:COMM:SERial commands).
- Contents of limit and data registers.
- Contents of math function and constant registers.
- Contents of the RAM disks.
- Calibration constants.
- Contents of the time capture buffer.
- The external keyboard setup.

**\*SRE****command/query**

Sets bits in the Service Request enable register.

Command Syntax:            \*SRE <number>|<bound>

                          <number> ::= a real number (NRf data)  
  limits: 0:255

                          <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"\*sre 40"  
                              OUTPUT 711;"\*Sre 17"

Query Syntax:             \*SRE?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: dependent on setting of \*PSC  
                              SCPI Compliance: confirmed

Description:

This command allows you to set bits in the Service Request enable register. Assign a decimal weight to each bit you want set (to 1) according to the following formula:

$$2^{(\text{bit\_number})}$$

with acceptable values for bit\_number being 0 through 7. Add the weights and then send the sum with this command.

---

**Note**                    The analyzer ignores the setting you specify for bit 6 of the Service Request enable register. This is because the corresponding bit of the Status Byte register is always enabled.

---

The analyzer requests service from the active controller when one of the following occurs:

- A bit in the Status Byte register changes from 0 to 1 while the corresponding bit of the Service Request enable register is set to 1.
- A bit in the Service Request enable register changes from 0 to 1 while the corresponding bit of the Status Byte register is set to 1.

The setting last specified with \*SRE is saved in nonvolatile memory. It can be recalled at power-up, depending on the setting of the Power-on Status Clear flag (set with \*PSC). When the flag is 0 at power-up, all bits in the Service Request enable register are set according to the saved \*SRE value. When the flag is 1 at power-up, all bits in the Service Request enable register are initialized to 0.

The query returns the current state of the Service Request enable register. The state is returned as a sum of the decimal weights of all set bits.

**\*STB?**

**query**

Reads the Status Byte register.

Query Syntax:               \*STB?

Example Statements:        OUTPUT 711;"\*Stb?"  
                              OUTPUT 711;"\*STB?"

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  variable  
                              SCPI Compliance:  confirmed

Description:

This command allows you to set bits in the Status Byte register. The state is returned as a sum of the decimal weights of all set bits. The decimal weight for each bit is assigned according to the following formula:

$$2^{(\text{bit\_number})}$$

with acceptable values for bit\_number being 0 through 7.

The register is not cleared by this query. To clear the Status Byte register, you must send the \*CLS command.

Bits in the Status Byte register are defined as follows:

- Bit 0 summarizes all enabled bits of the User Status register.
- Bit 1 is reserved.
- Bit 2 summarizes all enabled bits of the Device State register.
- Bit 3 summarizes all enabled bits of the Questionable Status register.
- Bit 4 is the Message Available (MAV) bit. It is set whenever there is something in the analyzer's output queue.
- Bit 5 summarizes all enabled bits of the Standard Event Status register.



- Bit 6, when read with this query (\*STB?), acts as the Master Summary Status (MSS) bit. It summarizes all enabled bits of the Status Byte register. (Bit 6 acts as the Request Service (RQS) bit when it is read by a serial poll.
- Bit 7 summarizes all enabled bits of the Operation Status register.

For more information on the Status Byte register, see chapter 1.

**\*TRG**

**command**

Triggers the analyzer when TRIG:SOUR is BUS.

Command Syntax:           \*TRG

Example Statements:        OUTPUT 711;"\*trg"  
                              OUTPUT 711;"\*Trg"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This command triggers the analyzer when the following two conditions are met:

- The GPIB is designated as the trigger source. (See the TRIG:SOUR BUS command.)
- The analyzer is waiting to trigger. (Bit 5 of the Operation Status register must be set). It is ignored at all other times.

The \*TRG command has the same effect as TRIG:IMM. It also has the same effect as the GPIB bus management command Group Execute Trigger (GET).

---

**\*TST?****query**

Tests the analyzer hardware and returns the results.

Query Syntax:            \*TST?

Example Statements:     OUTPUT 711;"\*TST?"  
                          OUTPUT 711;"\*tst?"

Return Format:           Integer

Attribute Summary:     Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

The analyzer's self-test performs a full calibration and then compares the calibration results to specified limits. If the results are within specified limits, the analyzer returns 0. If the results exceed the specified limits, the analyzer returns 1.

**\*WAI**

**command**

Holds off processing of subsequent commands until all preceding commands have been processed.

Command Syntax:            \*WAI

Example Statements:        OUTPUT 711;"\*Wai"  
                              OUTPUT 711;"\*WAI"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

Use \*WAI to hold off the processing of subsequent commands until all pending overlapped commands have been completed.

Some commands are processed sequentially by the analyzer. A sequential command holds off the processing of any subsequent commands until it has been completely processed. However, some commands do not hold off the processing of subsequent commands; they are referred to as overlapped commands. \*WAI ensures that overlapped commands are completely processed before subsequent commands (those sent after \*WAI) are processed.

See “Synchronization” in the GPIB Programmer’s Guide for additional information about the use of \*WAI and overlapped commands.

ABORt

---



## ABORt

---

This subsystem contains one command which aborts any measurement process.

---

**ABORt****command**

Stops the current measurement in progress.

Command Syntax:            ABORt

Example Statements:        OUTPUT 711;"abor"  
                              OUTPUT 711;"Abort"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This command aborts any measurement in progress and resets the trigger system. ABOR forces the trigger system to an idle state. ABORt forces the Measuring bit (bit 4) and the Averaging bit (bit 8) of the Operation Status register to 0. To restart a measurement you must send the INIT:IMM command.

The program message ABOR;:INIT:IMM serves a special synchronization function. When you send this message to restart a measurement, the analyzer's No Pending Operation (NPO) flag is set to 1 until the measurement is complete. The two commands that test the state of this flag—\*WAI and \*OPC—allow you to hold off subsequent actions until the measurement is complete. See chapter 3 of the GPIB Programmer's Guide for more information about synchronization.

---

**Note**                    When averaging is on ([SENSe:]AVERAge[:STATe] ON) the NPO flag is not set to 1 until *n* measurements have been combined into one trace. You specify the value of *n* with the [SENSe:] AVER:COUNT command.

---



ARM

---



## ARM

---

This subsystem contains commands that control the analyzer's trigger-arming functions. See the TRIGger subsystem for commands related to other triggering functions.

Figure 5-1 shows the model for the Agilent 35670A's ARM-INITiate-TRIGger functions.

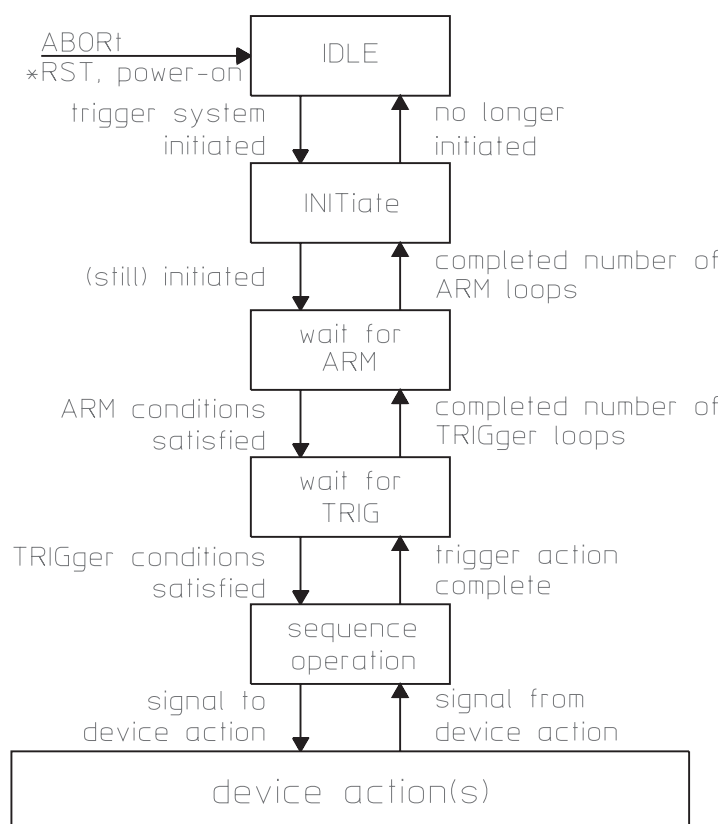


Figure 5-1. The Agilent 35670A's ARM-INIT-TRIG Functions

**ARM[:IMMEDIATE]****command**

Arms the trigger if ARM:SOUR is MAN.

Command Syntax:            ARM[:IMMEDIATE]

Example Statements:        OUTPUT 711;":ARM"  
                              OUTPUT 711;"arm:imm"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

Two conditions must be met before this command arms the trigger.

- Manual arming must be selected (ARM:SOUR is MAN).
- Bit 1 (Settling) or bit 6 (RDY for ARM) of the Operation Status condition register must be set to 1.

ARM:IMM is ignored at all other times.

For more information about how to use status registers, see “Programming the Status System” in the GPIB Programmer’s Guide .

**ARM:RPM:INCRement****command/query**

Specifies the number of RPM in a step for RPM step arming.

Command Syntax:            ARM:RPM:INCRement {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 1:500000

<unit> ::= [RPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"ARM:RMP:INCR 10"  
                              OUTPUT 711;"arm:rmp:increment 500"

Query Syntax:             ARM:RPM:INCRement?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +60 RPM  
                              SCPI Compliance: instrument-specific

Description:

This command determines the size of each step used when RPM step arming is enabled with the ARM:SOUR RPM command.

The first arm occurs when the RPM value reaches the threshold value (specified with the ARM:RPM:THR command). Subsequent arms occur at the RPM interval specified with this command.

**ARM:RPM:MODE****command/query**

Enables the Start RPM Arming qualifier.

Command Syntax:            ARM:RPM:MODE OFF | 0 | UP | DOWN

Example Statements:        OUTPUT 711; "Arm:Rpm:Mode UP"  
                              OUTPUT 711; "ARM:RPM:MODE DOWN"

Query Syntax:              ARM:RPM:MODE?

Return Format:             OFF | UP | DOWN

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: UP  
                              SCPI Compliance: instrument-specific

**Description:**

For a runup measurement send ARM:RPM:MODE UP. The first arm occurs when the RPM value reaches the specified threshold value (specified with ARM:RPM:THR).

For a rundown measurement send ARM:RPM:MODE DOWN. The first arm occurs when the RPM value reaches the specified threshold value (specified with ARM:RPM:THR).

To specify subsequent arms to occur at RPM intervals measured from the starting RPM value see the ARM:TIMER command.

In order analysis instrument mode (INST:SEL ORD; Option 1D1), you can disable the Start RPM Arming qualifier. Send ARM:RPM:MODE OFF.

**ARM:RPM:THReshold****command/query**

Specifies the starting RPM value.

Command Syntax:           ARM:RPM:THReshold {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)

          limits: 5:491520

<unit> ::= [RPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"ARM:RPM:THR 1000"  
                          OUTPUT 711;"arm:rpm:threshold 8000"

Query Syntax:             ARM:RPM:THReshold?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +600 RPM  
                          SCPI Compliance: instrument-specific

Description:

This command is only valid when the Start RPM Arming qualifier is enabled with the ARM:RPM:MODE command.

The starting value is either the lowest RPM used for devices with increasing RPM (ARM:RPM:MODE UP) or the highest RPM for devices with decreasing RPM (ARM:RMP:MODE DOWN).

**ARM:SOURce****command/query**

Specifies the type of arming for the analyzer's trigger.

Command Syntax:            ARM:SOURce IMMEDIATE|MANual|RPM|TIMer

Example Statements:        OUTPUT 711;":arm:sour MANUAL"  
                              OUTPUT 711;"Arm:Sour MANUAL"

Query Syntax:              ARM:SOURce?

Return Format:             IMM|MAN|RPM|TIM

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:   IMM  
                              SCPI Compliance:  confirmed

**Description:**

To select automatic arming send IMM. The analyzer waits for the hardware to settle and then waits for a trigger signal (specified by TRIG:SOUR) before starting the measurement. When the measurement is completed, the trigger is automatically re-armed.

To select manual arming send MAN. The analyzer waits for the hardware to settle then waits for the ARM[:IMM] command, and then waits for a trigger signal before starting the measurement. The ARM[:IMM] command must be sent to re-arm the trigger after the measurement is completed.

To select RPM step arming send RPM. The analyzer waits for the hardware to settle, waits for the next RPM level to be reached, and then waits for the trigger signal before starting the measurement. See the TRIGger:TACHometer commands for more information about using the analyzer's tachometer.

To select time step arming send TIM. The instrument waits for the hardware to settle, waits for a trigger and starts the measurement by collecting the first time record. The analyzer then waits until the end of the time step interval (specified by ARM:TIM) and waits for the trigger signal before collecting the next time record. In waterfall displays, all waterfall traces are referenced from the start time. See figure 5-2.



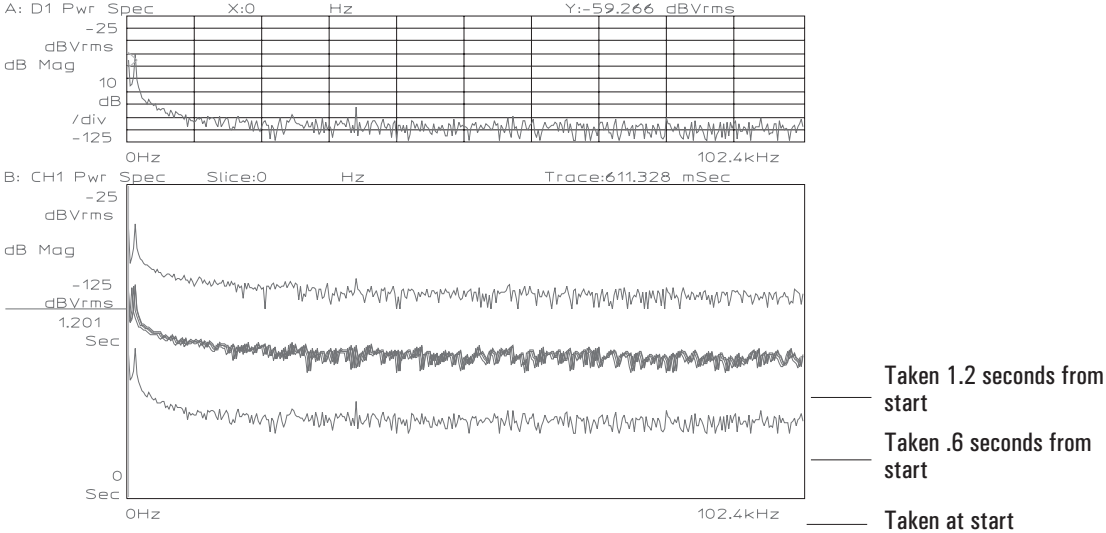


Figure 5-2. Recorded Time of Triggered Event Referenced to Start Time

The [SENSE:]AVERAge:TCONtrol REPEAT command allows you to get a complete set of averages at each arming event during manual, RPM step and time step arming.

**ARM:TIMer****command/query**

Specifies the size of the step used in time step arming.

Command Syntax:            ARM:TIMer {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: 0.0:500000

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"arm:tim 3600"  
                          OUTPUT 711;"ARM:TIMER 1.5"

Query Syntax:             ARM:TIMer?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.5 S  
                          SCPI Compliance: confirmed

Description:

This command is only valid with time step arming (ARM:SOUR TIM).

The default unit is seconds.

## CALCulate

---



## CALCulate

---

This subsystem contains commands that control the processing of measurement data. The block diagram figure 6-1 shows you how measurement data is processed.

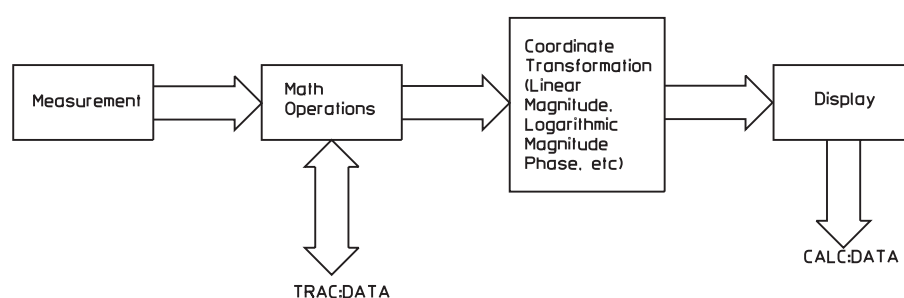


Figure 6-1. Flow of Measurement Data in the Agilent 35670A

After measurement data is collected, any specified math operations are performed. Data is then transformed into the specified coordinate system and sent to the display. TRAC:DATA gives you access to the raw measurement data after the analyzer performs math operations. CALC:DATA gives you access to the data—after the coordinate transformation.

The CALCulate subsystem lets you:

- Specify the type of measurement data.
- Select a coordinate system for display of the measurement data.
- Define trace math functions and constants.
- Curve fit and synthesize trace data.
- Perform limit testing.
- Control the analyzer's marker functions.
- Read marker values.
- Transfer coordinate transformed data to your controller.

---

**Note** You can transfer measurement data from the analyzer with either the TRAC:DATA command or the CALC:DATA command. However, you can only transfer data to the analyzer with the TRAC:DATA command.

---

The CALCulate mnemonic contains an optional trace specifier: [1|2|3|4]. To direct a command to trace A, omit the specifier or use 1. To direct a command to trace B, use 2; to trace C, use 3; and to trace D, use 4. Commands that are not trace-specific—the CALC:MATH commands, the CALC:CFIT commands and the CALC:SYNT commands—ignore the specifier.

**CALCulate[1 | 2 | 3 | 4]:ACTive****command/query**

Selects the active trace(s).

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :ACTive A | B | C | D | AB | CD | ABCD

Example Statements:        OUTPUT 711; "CALC4:ACTIVE C"  
                              OUTPUT 711; "calc4:active ABCD"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :ACTive?

Return Format:             A | B | C | D | AB | CD | ABCD

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: trace A  
                              SCPI Compliance: instrument-specific

Description:

In addition to the individual traces, you can set multiple traces to be active at any time.

The trace pairs of A/B or C/D can be active as well as all four traces—A/B/C/D. If A/B or A/B/C/D are active, trace A is considered the “most active trace.” If traces C/D are active, trace C is considered the “most active trace.”

The trace pairs of A/C or B/D *cannot* be active at the same time.

The trace specifier, CALCULATE[1|2|3|4], is not used.

**CALCulate[1 | 2 | 3 | 4]:CFIT:ABORt**

**command**

Aborts the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:ABORt

Example Statements:        OUTPUT 711;":Calc3:Cfit:Abort"  
                              OUTPUT 711;"CALC:CFIT:ABOR"

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

This command aborts the current curve fit operation. As a result of the CALC:CFIT:ABOR command, the curve fit table is not updated.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---



---

**CALCulate[1 | 2 | 3 | 4]:CFIT:COPY****command**

Copies the synthesis table into the curve-fit table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:COPY SYNThesis

Example Statements:        OUTPUT 711;"calc:cfit:copy SYNTHESIS"  
                              OUTPUT 711;"Calculate:Cfit:Copy SYNTHESIS"

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The synthesis table must be in pole-zero format. To convert a synthesis table to pole-zero format, use the CALC:SYNT:TTYP PZER command.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:DATA****command/query**

Loads values into the curve fit table.

Command Syntax:            CALCulate:CFIT:DATA &lt;BLOCK&gt;

&lt;BLOCK&gt; ::= see Description

Example Statements:        OUTPUT 711;":CALCULATE:CFIT:DATA BLOCK"  
                          OUTPUT 711;"calc:cfit:data BLOCK"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:DATA?

Return Format:             DEF\_BLOCK

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

This command transfers a complete curve fit table from your controller to the analyzer.

When you transfer a curve fit table to the analyzer, you must use the definite length block syntax. Data must be 64-bit binary floating-point numbers (see the FORMat[:DATA] REAL command). The elements of the definite length block for a curve fit table are defined below.

The first point specifies the table type; followed by three points which specify the size of the table. The next 80 points are the complex pairs. The remaining points are descriptors. Points 152 - 174 indicate which terms are fixed. Points 173 - 175 are adjusters; gain, frequency scale and time delay.

See "Block Data" in the GPIB Programmer's Guide for additional information about transferring block data.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

```

<BLOCK> ::= #41400<Point1><Point2> . . . <Point175>
<Point1> ::= Table type
           <0> = pole zero
           <1> = pole residue
           <2> = polynomial
<Point2> ::= number_of_lines_in_left_column
<Point3> ::= number_of_lines_in_right_column
<Point4> ::= number_of_lines_in_Laurent_column
<Point5> ::= real_part_first_term_in_left_column
<Point6> ::= imaginary_part_first_term_in_left_column
<Point7> ::= real_part_second_term_in_left_column
<Point8> ::= imaginary_part_second_term_in_left_column
.
.
.
<Point47> ::= real_part_first_term_in_right_column
<Point48> ::= imaginary_part_first_term_in_right_column
.
.
.
<Point89> ::= real_part_first_term_in_Laurent_column
<Point90> ::= imaginary_part_first_term_in_Laurent_column
.
.
.
<Point131> ::= first_curve_fit_term_left_column
<Point132> ::= second_curve_fit_term_left_column
<curve_fit_term> ::= 0 for moveable
                   1 for fixed
.
.
.
<Point152> ::= first_curve_fit_term_right_column
<Point153> ::= second_curve_fit_term_right_column
<curve_fit_term> ::= 0 for moveable
                   1 for fixed
.
.
.
<Point173> ::= gain
<Point174> ::= frequency_scale
<Point 175> ::= time_delay

```

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:DESTination****command/query**

Selects the data register for the results of the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:DESTination D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Example Statements:        OUTPUT 711;"Calc4:Cfit:Dest D2"  
                              OUTPUT 711;"CALC:CFIT:DEST D4"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:DESTination?

Return Format:             D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: D6  
                              SCPI Compliance: instrument-specific

**Description:**

This command specifies which data register holds the synthesis of the intermediate and final curve fit models. The default register is D6.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:FREQuency[:AUTO]****command/query**

Specifies the region included in the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:FREQuency [:AUTO] OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;":calculate2:cfit:freq ON"  
                              OUTPUT 711;"Calculate:Cfit:Frequency:Auto OFF"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:FREQuency [:AUTO] ?

Return Format:             Integer

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: ON (+1)  
                              SCPI Compliance: instrument-specific

Description:

To specify the full span send CALC:CFIT:FREQ ON.

To limit the region of the curve fit send CALC:CFIT:FREQ OFF. The commands, CALC:CFIT:FREQ:STAR and CALC:CFIT:FREQ:STOP define the region of a curve fit over a limited span.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---



**CALCulate[1 | 2 | 3 | 4]:CFIT:FREQUENCY:STOP****command/query**

Specifies the stop frequency for a curve fit operation over a limited frequency span.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:FREQUENCY:STOP {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRF data  
                  limits: 0.390625:115000.0

<unit> ::= [HZ]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :Calculate:Cfit:Freq:Stop 1.2 khz"  
                          OUTPUT 711;"CALC:CFIT:FREQ:STOP 800"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :CFIT:FREQUENCY:STOP?

Return Format:            Real

Attribute Summary:       Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: +5.12E+004  
                          SCPI Compliance: instrument-specific

Description:

This command is not valid unless the CALC:CFIT:FREQ OFF command has been sent.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:FSCale****command/query**

Specifies the frequency scaling used in the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:FSCale <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 1e-6:1e6  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;"calculate2:cfit:fscale 604144"  
                          OUTPUT 711;"Calc2:Cfit:Fscale 582699"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:FSCale?

Return Format:            Real

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: +1.0  
                          SCPI Compliance: instrument-specific

Description:

The analyzer scales the frequency axis (the X-axis) by f/frequency scale, where f is frequency in Hertz.

---

**Note**                    This command must be sent before CALC:CFIT. This command is not trace specific. It ignores the trace specifier.

---



---

**CALCulate[1 | 2 | 3 | 4]:CFIT[:IMMEDIATE]****command**

Starts the curve fit process.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT[:IMMEDIATE]

Example Statements:        OUTPUT 711;":CALC3:CFIT"  
                              OUTPUT 711;"calc:cfit:immediate"

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

The results of the curve fit are stored in the curve fit data register specified by the CALC:CFIT:DEST command.

To abort the curve fit operation, send CALC:CFIT:ABOR.

---

**Note**                   This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:ORDer:AUTO****command/query**

Determines the operation of the curve fitter.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:ORDer:AUTO OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;"Calc:Cfit:Ord:Auto OFF"  
                              OUTPUT 711;"CALCULATE3:CFIT:ORDER:AUTO OFF"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:ORDer:AUTO?

Return Format:             Integer

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: ON (+1)  
                              SCPI Compliance: instrument-specific

**Description:**

This command determines the curve fit operation mode. AUTO ON (the default value) places the curve fitter in automatic order selection. AUTO OFF gives a model with fixed numerator and denominator order.

In automatic order selection (CALC:CFIT:ORDER:AUTO is ON), the curve fit operation starts with 1 pole and 1 zero. If the fit is poor, the orders increment and another curve fit operation is performed. This iterative process continues until a model is found which more closely matches the measured frequency response. The curve fit operation uses the number of poles and zeros specified with the CALC:CFIT:ORD:POLES and CALC:CFIT:ORD:ZER commands as the upper bounds for this iterative search.

If CALC:CFIT:ORDER:AUTO OFF is sent, the curve fit operation provides a model with the number of poles and zeros specified with the CALC:CFIT:ORD:POLES and CALC:CFIT:ORD:ZER commands. There is no iterative search for a better model.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:ORDer:POLes****command/query**

Specifies the number of poles used in the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:ORDer:POLes <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:20

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":calc:cfit:ord:poles 5"  
                          OUTPUT 711;"Calc:Cfit:Order:Pol 14"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :CFIT:ORDer:POLes?

Return Format:            Integer

Attribute Summary:       Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: +20  
                          SCPI Compliance: instrument-specific

**Description:**

The actual number of poles used in the curve fit operation is determined by the mode selected by the CALC:CFIT:ORD:AUTO command.

If CALC:CFIT:ORD:AUTO is ON (the default value), an optimum number of poles is used for the model. The number specified with this command represents the upper bounds for the iterative search. The number of poles will not exceed the number specified by CALC:CFIT:ORD:POL.

If CALC:CFIT:ORD:AUTO OFF is sent, the number of poles specified by CALC:CFIT:ORD:POL is used for the model.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---



**CALCulate[1 | 2 | 3 | 4]:CFIT:TDElay****command/query**

Specifies a time delay value for the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:TDElay {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: -100:100

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :Calculate4:Cfit:Tdelay -19.0035"  
                          OUTPUT 711;"CALC:CFIT:TDELAY 71.4756"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :CFIT:TDElay?

Return Format:            Real

Attribute Summary:       Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: +0  
                          SCPI Compliance: instrument-specific

Description:

Use this command to include a time delay value which removes any time delay from the frequency response to be fitted.

Positive delay is entered as a positive value.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:WEIGHT:AUTO****command/query**

Determines the weighting function used in the curve fit operation.

Command Syntax:            CALCulate [1 | 2 | 3 | 4] :CFIT:WEIGHT:AUTO OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;"calc4:cfit:weig:auto ON"  
                              OUTPUT 711;"Calculate:Cfit:Weight:Auto ON"

Query Syntax:              CALCulate [1 | 2 | 3 | 4] :CFIT:WEIGHT:AUTO?

Return Format:             Integer

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: ON (+1)  
                              SCPI Compliance: instrument-specific

**Description:**

If CALC:CFIT:WEIGHT:AUTO is ON (the default value), the curve fit operation automatically generates a weighting function. It stores the result in the curve fit weight register. See CALC:CFIT:WEIG:REG command for more information about the curve fit weight register.

If CALC:CFIT:WEIGHT:AUTO OFF is sent, the curve fit operation uses the weighting function stored in the specified register. The curve fit operation will abort if the curve fit weight register is empty or contains invalid data.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

**CALCulate[1 | 2 | 3 | 4]:CFIT:WEIGht:REGister****command/query**

Selects the data register which contains the weighting function for the curve fit operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :CFIT:WEIGht:REGister  
D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Example Statements:        OUTPUT 711; ":CALC:CFIT:WEIG:REGISTER D7"  
OUTPUT 711; "calc:cfit:weight:reg D1"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :CFIT:WEIGht:REGister?

Return Format:             D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
Synchronization Required: no  
Preset State:   D7  
SCPI Compliance:  instrument-specific

Description:

The default register is D7.

If you select the auto weight feature (the default is CALC:CFIT:WEIG:AUTO ON), the analyzer automatically generates a weighting function and stores it in the register specified with this command.

If you disable the auto weight feature by sending the CALC:CFIT:WEIG:AUTO OFF command, the curve fit operation uses the weighting function stored in the register specified with this command. The analyzer does not generate and store a new weighting function.

The curve fit operation will abort if the curve fit weight register is empty or contains invalid data.

---

**Note**                   This command is not trace specific. It ignores the trace specifier.

---

**CALCulate[1 | 2 | 3 | 4]:DATA?****query**

Returns trace data that has been transformed to the currently selected coordinate transform (specified with the CALC:FORMat command).

Query Syntax:                CALCulate [1 | 2 | 3 | 4] :DATA?

Example Statements:        OUTPUT 711;"Calculate:Data?"  
                               OUTPUT 711;"CALC4:DATA?"

Return Format:              <DEF\_BLOCK>

If FORMat [:DATA] REAL:

```
<DEF_BLOCK> ::= #<byte><length_bytes> <1st_Y-axis_value> . . .
               <last_Y-axis_value>
               <byte> ::= number of length bytes to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)
```

If FORMat[:DATA] ASCii:

```
<DEF_BLOCK> ::= <1st_Y-axis_value> . . .<last_Y-axis_value>
```

Attribute Summary:        Option:    not applicable  
                               Synchronization Required: no  
                               Preset State:    not applicable  
                               SCPI Compliance: confirmed

Description:

This query returns a definite length block of coordinate-transformed trace data.

The trace specifier CALCulate [1|2|3|4] determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

The block is returned as a series of amplitude (Y-axis) values. The unit for these values is the same as the reference level unit. To determine the unit, send DISP:TRAC:Y:BOTT? UNIT.

For orbit diagrams (CALC:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]') the block is a series of real-value pairs (<1st\_X-axis\_value>, <1st\_Y-axis\_1\_value>, . . . <last\_X-axis\_value>, <last\_Y-axis\_value>).

For Nyquist diagrams (CALC:FORM NYQ) the block is a series of real and imaginary pairs (<1st\_real\_value>, <1st\_imaginary\_value>, . . . <last\_real\_value>, <last\_imaginary\_value>).

For polar diagrams (CALC:FORM POL) the block is a series of magnitude and phase pairs (<1st\_magnitude\_value>, <1st\_phase\_value>, . . . <last\_magnitude\_value>, <last\_phase\_value>).

In octave analysis instrument mode (INST:SEL OCT;Option 1D1) the second to the last value in the data block is the weighted overall band. The last value in the data block is the overall band. These values are included in the data block, even if they are not displayed.



See table 6-1 to determine the number of values (display points) with variable resolution. The CALC:DATA:HEAD:POIN? query returns the number of values along the specified display's X-axis. Refer to the CALC:X:DATA? query for information on retrieving X-axis values.

This query has no command form. Therefore, you cannot return trace data to the display with CALC:DATA. To send data that has not been transformed, use the TRAC:DATA command. See the introduction to this chapter for more information about the differences between these commands.

For more information about transferring block data, see Appendix F, "Example Programs," or the GPIB Programmer's Guide .

Table 6-1. Number of displayed data points in variable resolution

FFT Instrument Mode

Resolution	Baseband		Zoom	
	Number of frequency points (frequency data)	Number of real time points (time data)	Number of frequency points (frequency data)	Number of complex time points (real/imaginary pairs) (time data)
100	101	256	101	128
200	201	512	201	256
400	401	1024	401	512
800	801	2048	801	1024

Correlation Instrument Mode  
(no complex data)

Resolution	Auto- and cross correlation (real data)			Time domain (real data)
	0 to T/2	-T/2 to T/2	-T/4 to T/4	
100	128	256	128	256
200	256	512	256	512
400	512	1024	512	1024
800	1024	2048	1024	2048

**CALCulate[1 | 2 | 3 | 4]:DATA:HEADer:POINts?****query**

Returns the number of values in the data block returned with the CALC:DATA? query.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :DATA:HEADer:POINts?

Example Statements:            OUTPUT 711;":calc:data:head:poin?"  
                                  OUTPUT 711;"Calculate:Data:Header:Poin?"

Return Format:                 Integer

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required:  no  
                                  Preset State:  not applicable  
                                  SCPI Compliance:  instrument-specific

**Description:**

The display's X-axis is divided into discrete points. Use this query to determine how many discrete points there are along the specified display's X-axis. This is the number of values sent to the analyzer's output queue when you send the CALC:DATA? query.

For orbit diagrams (CALC:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]'), Nyquist diagrams (CALC:FORM NYQ), and polar diagrams (CALC:FORM POL), the number of points represents the number of *pairs of values*, not the number of discrete values.

In octave analysis instrument mode (INST:SEL OCT;Option 1D1) the number of points includes the weighted overall band and the overall band.

**CALCulate[1 | 2 | 3 | 4]:FEED****command/query**

Selects the measurement data to be displayed in the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :FEED <CMDSTR>

<CMDSTR> ::= command string  
              see description

Example Statements:       OUTPUT 711;"CALC:FEED `XFR:POW 1`"  
                          OUTPUT 711;"calculate:feed `XVOL:VOLT 1,2`"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :FEED?

Return Format:             STRING

Attribute Summary:       Option: not applicable  
                          Synchronization Required: yes  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

## Description:

This command selects the measurement results. The available measurement data varies for different instrument modes. See table 6-2 for a complete listing of measurement results and their related <CMDSTR> for each instrument mode.

The trace specifier CALCulate [1|2|3|4] determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D. If you omit the channel specifier, it defaults to channel 1.

In FFT analysis instrument mode (INST:SEL FFT), the following commands are available:

To select the contents of the specified data register, send CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'.

To select the contents of the time capture buffer for the specified channel, send CALC[1|2|3|4]:FEED 'TCAP [1|2|3|4]'.

To select the contents of the specified waterfall register, send CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'.

To select the linear spectrum function for the specified channel, send CALC[1|2|3|4]:FEED 'XFR:POW:LIN [1|2|3|4]'.

To select the power spectrum function for the specified channel, send CALC[1|2|3|4]:FEED 'XFR:POW [1|2|3|4]'.

## CALCulate

To select the coherence function, send `CALC[1|2|3|4]:FEED 'XFR:POW:COH [1,2|1,3|1,4|3,4]'`. The analyzer must be in 2 channel instrument mode (INPut 2 ON) for selection 1,2. INPut 4 must be ON for selections 1,3 and 1,4. INPut 4 must be ON and SENSe:REFerence must be PAIR for selection 3,4.

To select the cross spectrum, send `CALC[1|2|3|4]:FEED 'XFR:POW:CROS [1,2|1,3|1,4|3,4]'`. The analyzer must be in 2 channel instrument mode (INPut 2 ON) for selection 1,2. INPut 4 must be ON for selections 1,3 and 1,4. INPut 4 must be ON and SENSe:REFerence must be PAIR for selection 3,4.

To select the most recent frequency response function, send `CALC[1|2|3|4]:FEED 'XFR:POW:RAT [2,1|3,1|4,1|4,3]'`. The analyzer must be in 2 channel instrument mode (INPut 2 ON) for selection 2,1. INPut 4 must be ON for selections 3,1 and 4,1. INPut 4 must be ON and SENSe:REFerence must be PAIR for selection 4,3.

To select the most recent time record for the specified channel, send `CALC[1|2|3|4]:FEED 'XTIM:VOLT [1|2|3|4]'`.

To select the most recent windowed time record for the specified channel, send `CALC[1|2|3|4]:FEED 'XTIM:VOLT:WIND [1|2|3|4]'`.

To select the orbit diagram, send `CALC[1|2|3|4]:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]'`. The orbit diagram presents channel 1 time along the X-axis and channel 2, channel 3, or channel 4 time along the Y- axis. The analyzer must be in 2 channel instrument mode (INPut 2 ON) for selection 1,2. INPut 4 must be ON for selections 1,3 and 1,4. INPut 4 must be ON and SENSe:REFerence must be PAIR for selection 3,4.

In correlation instrument mode (INST:SEL CORR), the following commands are available:

`CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'`.

`CALC[1|2|3|4]:FEED 'TCAP [1|2|3|4]'`.

`CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'`.

`CALC[1|2|3|4]:FEED 'XTIM:VOLT [1|2|3|4]'`.

`CALC[1|2|3|4]:FEED 'XTIM:VOLT:WIND [1|2|3|4]'`.

To select the most recent autocorrelation for the specified channel, send `CALC[1|2|3|4]:FEED 'XTIM:CORR [1|2|3|4]'`.

To select the most recent cross correlation for the specified channel, send `CALC[1|2|3|4]:FEED 'XTIM:CORR:CROS [1,2|1,3|1,4|3,4]'`. The analyzer must be in 2 channel instrument mode (INPut 2 ON) for selection 1,2. INPut 4 must be ON for selections 1,3 and 1,4. INPut 4 must be ON and SENSe:REFerence must be PAIR for selection 3,4.

In histogram instrument mode (INST:SEL HIST) :

CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'.

CALC[1|2|3|4]:FEED 'TCAP [1|2|3|4]'.

CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'.

To select the most recent unfiltered time record for the specified channel, send  
CALC[1|2|3|4]:FEED 'XTIM:VOLT [1|2|3|4]'.

To select the most recent histogram for the specified channel, send CALC[1|2|3|4]:FEED  
'XTIM:VOLT:HIST [1|2|3|4]'.

To select the probability density function for the specified channel, send CALC[1|2|3|4]:FEED  
'XTIM:VOLT:PDF [1|2|3|4]'. The histogram is normalized to unit area.

To select the cumulative density function for the specified channel, send CALC[1|2|3|4]:FEED  
'XTIM:VOLT:CDF [1|2|3|4]'. This shows the probability that a level ® to a specific level occurred.

In order analysis instrument mode (INST:SEL ORD; Option 1D0), available commands are:

CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'.

CALC[1|2|3|4]:FEED 'TCAP [1|2|3|4]'.

CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'.

CALC[1|2|3|4]:FEED 'XFR:POW [1|2|3|4]'. Order track must be off (sent with the  
ORD:TRAC:STAT OFF command).

CALC[1|2|3|4]:FEED 'XTIM:VOLT [1|2|3|4]'. Order track must be off (sent with the  
ORD:TRAC:STAT OFF command).

CALC[1|2|3|4]:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]'. Order track must be off (sent with the  
ORD:TRAC:STAT OFF command). The analyzer must be in 2 channel instrument mode (INPut 2  
ON) for selection 1,2. INPut 4 must be ON for selections 1,3 and 1,4. INPut 4 must be ON and  
SENse:REfERENCE must be PAIR for selection 3,4.

To display one of five order tracks, send CALC:FEED 'XORD:TRAC [1|2|3|4],[1|2|3|4|5]'. The  
first parameter specifies the channel. The second parameter specifies the order track. Order track  
must be on (sent with the ORD:TRAC:STAT ON command).

To display time versus RPM (which tells you how long the RPM runup or rundown took), send  
CALC:FEED 'XRPM:PROF'. Order track must be on (sent with the ORD: TRAC:STAT ON  
command).

## CALCulate

To select composite power for the specified channel, send `CALC[1|2|3|4]:FEED 'XFR:POW:COMP [1|2|3|4]'`. Order track must be on (sent with the `ORD:TRAC:STAT ON` command). The `[SENSe:]WINDow:ORDer:DC` command specifies whether the analyzer should use the dc bins in calculating composite power.

In octave analysis instrument mode (`INST:SEL OCT`; Option 1D1), available commands are:

`CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'`.

`CALC[1|2|3|4]:FEED 'TCAP [1|2|3|4]'`.

`CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'`.

`CALC[1|2|3|4]:FEED 'XFR:POW [1|2|3|4]'`.

In swept sine instrument mode (`INST:SEL SINE`; Option 1D2), available commands are:

`CALC[1|2|3|4]:FEED 'D1|D2|D3|D4|D5|D6|D7|D8'`.

`CALC[1|2|3|4]:FEED 'W1|W2|W3|W4|W5|W6|W7|W8'`.

`CALC[1|2|3|4]:FEED 'XFR:POW:LIN [1|2|3|4]'`.

`CALC[1|2|3|4]:FEED 'XTIM:VOLT [1|2|3|4]'`.

`CALC[1|2|3|4]:FEED 'XFR:POW:CROS [1,2|1,3|1,4|3,4]'`. The analyzer must be in 2 channel instrument mode (`INPut 2 ON`) for selection 1,2. `INPut 4` must be ON for selections 1,3 and 1,4. `INPut 4` must be ON and `SENSe:REFerence` must be PAIR for selection 3,4.

`CALC[1|2|3|4]:FEED 'XFR:POW:RAT [2,1|3,1|4,1|4,3]'`. The analyzer must be in 2 channel instrument mode (`INPut 2 ON`) for selection 2,1. `INPut 4` must be ON for selections 3,1 and 4,1. `INPut 4` must be ON and `SENSe:REFerence` must be PAIR for selection 4,3.

To display the normalized variance for the specified channel, send `CALC[1|2|3|4]:FEED 'XFR:POW:VAR [1|2|3|4]'`.

Table 6-2. Measurement Results for each Instrument Mode

Measumrent Data CALC:FEED command	INSTrument:SElect					
	FFT	CORR	HIST	ORD	OCT	SINE
Auto correlation XTIM:CORR [1   2   3   4]		X				
Capture buffer TCAP [1   2   3   4]	X	X	X	X	X	
Coherence XFR:POW:COH [1,2   1,3   1,4   3,4]	X					
Composite Power XFR:POW:COMP [1   2   3   4]				X		
Cross Correlation XTIM:CORR:CROS [1,2   1,3   1,4   3,4]		X				
Cross Spectrum XFR:POW:CROS [1,2   1,3   1,4   3,4]	X					X
Cumulative Density Function XTIM:VOLT:CDF [1   2   3   4]			X			
Data Register [D1   D2   D3   D4   D5   D6   D7   D8]	X	X	X	X	X	X
Frequence Response XFR:POW:RAT [2,1   3,1   4,1   4,3]	X					X
Histogram XTIM:VOLT:HIST [1   2   3   4]			X			
Linear Spectrum XFR:POW:LIN [1   2   3   4]	X					X
Normalized Variance XFR:POW:VAR [1   2   3   4]						X
Orbit Diagram XVOL:VOLT [1,2   1,3   1,4   3,4]	X			X		
Order Track XORD:TRACK [1   2   3   4],[1   2   3   4   5]				X		
Power Spectrum XFR:POW [1   2   3   4]	X			X	X	
Probability Density Function XTIM:VOLT:PDF [1   2   3   4]			X			
RPM Profile XRPM:PROF				X		
Time XTIM:VOLT [1   2   3   4]	X	X		X		X
Unfiltered Time XTIM:VOLT [1   2   3   4]			X			

Table 6-2. Measurement Results for each Instrument Mode (continued)

Measument Data CALC:FEED command	INSTrument:SElect					
	FFT	CORR	HIST	ORD	OCT	SINE
Waterfall Register [W1   W2   W3   W4   W5   W6   W7   W8]	X	X	X	X	X	X
Windowed Time XTIM:VOLT:WIND [1   2   3   4]	X	X				



**CALCulate[1 | 2 | 3 | 4]:FORMat****command/query**

Selects a coordinate system for displaying measurement data and for transferring coordinate transformed data to a controller.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :FORMat  
 MLINear | MLOGarithmic | PHASe | REAL | IMAGinary | NYQuist | UPHase  
 | GDElay | POLar

Example Statements:        OUTPUT 711;"CALC3:FORMAT IMAGINARY"  
 OUTPUT 711;"calc4:format IMAGINARY"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :FORMat?

Return Format:             MLIN | MLOG | PHAS | REAL | IMAG | NYQ | UPH | GDEL | POL

Attribute Summary:        Option:   not applicable  
 Synchronization Required:  yes  
 Preset State:   MLOG trace A and trace C  
                   REAL trace B and trace D  
 SCPI Compliance:  confirmed

## Description:

To select a coordinate system that displays linear magnitude data along the Y-axis, send CALC:FORM MLIN.

To display linear magnitude data on a logarithmic Y-axis scale, send CALC:FORM MLIN::DISP:TRAC:Y:SPAC LOG.

To select a coordinate system that displays logarithmic magnitude data on a linear Y-axis scale, send CALC:FORM MLOG.

To select a coordinate system that displays wrapped phase along the Y-axis, send CALC:FORM PHAS. Phase wraps at the display boundaries. If you change the display boundaries with the DISP:WIND:TRAC:Y:TOP and DISP:WIND:TRAC:Y:BOTTOM commands, you change where phase wraps on the display.

To select a coordinate system that displays imaginary numbers along the Y-axis, send CALC:FORM IMAG. This coordinate system shows the imaginary component of complex data at each point along the X-axis. If the data point is real rather than complex, a value of 0 is displayed for all X-axis points.

To select a coordinate system that displays real numbers along the Y-axis, send CALC:FORM REAL. This coordinate system shows real data or the real component of complex data at each point along the X-axis. This and the polar diagram (CALC:FORM POLar) are the only valid selection for orbit displays (CALC:FEED 'XVOL:VOLT 1,2'). CALC:FORM REAL is the default selection for orbit displays.

## CALCulate

To select a coordinate system that displays unwrapped phase along the Y-axis, send CALC:FORM UPH. The displayed phase is referenced to the lowest measured frequency. There is no phase wrapping.

To select a coordinate system that displays a Nyquist diagram (imaginary numbers along the Y-axis and real numbers along the X-axis), send CALC:FORM NYQ. It is not valid for RPM Profiles (CALC:FEED 'XRPM:PROF') and orbit diagrams (CALC:FEED 'XVOL:VOLT').

To select a coordinate system that displays a polar diagram (magnitude and phase displayed as a rotating vector), send CALC:FORM POL. It is not valid for RPM profiles (CALC:FEED 'XRPM:PROF').

To select a coordinate system that displays time on the Y-axis and frequency on the X-axis, send CALC:FORM GDEL. Group delay is defined as the negative of the derivative of the phase response. It shows phase delays in time rather than degrees of phase shift.

$$\tau_g = -d\phi/d\omega$$

Where

$\phi$  = phase response in radians

$\omega$  = frequency in radians/second

The analyzer uses a smoothing aperture to define the resolution of the group delay display. To specify the smoothing aperture send the CALC:GDAPerture:APERture command. Group delay is not valid for time data (CALC:FEED 'XTIM:VOLT'). It is normally used in frequency response measurements.

The trace specifier, CALCulate [1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D. Use the query, CALC:FORMat?, to determine the current coordinate system for the selected trace.

To query display data—that is, the analyzer has applied a coordinate system to the measurement data—use the CALC:DATA command. To access “raw” measurement data—only math operations have been performed on the data—use the TRAC:DATA command.

**CALCulate[1 | 2 | 3 | 4]:GDAPerture:APERture****command/query**

Specifies the phase-smoothing aperture used with group delay trace coordinates.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :GDAPerture:APERture {<number> [<unit>] } |<step> |<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:20

<unit> ::= [PCT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"CALC:GDAP:APER 4"  
                              OUTPUT 711;"calculate3:gdaperture:aper 5.0 PCT"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :GDAPerture:APERture?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: 2.0 PCT  
                              SCPI Compliance: confirmed

## Description:

This command specifies the phase-smoothing aperture used with the CALC:FORMat GPDElay command. The greater the aperture value, the greater the smoothing effect on the displayed data.

The aperture value is specified as a percentage. It is the ratio of the desired aperture span to the measured frequency span. The analyzer rounds the aperture value to the nearest frequency bin which is dependent upon the number of lines of resolution set with the SENSE:FREQUENCY:RESOLUTION command. See table 6-3 for the step size of aperture values in relation to the analyzer's resolution.

---

**Note**      CALC:GDAPERTURE:APERTURE is set to the minimum value after a reset (\*RST).  
See the table below.

---

Table 6-3. Lines of Resolution versus Aperture Step Size

Resolution	Aperture Step Size (in percent)
100	1.0 PCT
200	0.5 PCT
400	0.25 PCT
800	0.125 PCT

**CALCulate[1 | 2 | 3 | 4]:LIMit:BEEP[:STATe]****command/query**

Turns the limit-fail beeper on and off.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:BEEP [:STATe] OFF | 0 | ON | 1

Example Statements:        OUTPUT 711; ":Calc2:Lim:Beep ON"  
                               OUTPUT 711; "CALC:LIMIT:BEEP:STAT OFF"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :LIMit:BEEP [:STATe] ?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: OFF (+0)  
                               SCPI Compliance: instrument-specific

Description:

The limit-fail beeper emits an audible tone when all of the following conditions are met:

- CALC:LIM:BEEP is ON.
- CALC:LIM:STAT is ON.
- The trace falls outside its current limits.

You can use CALC:LIM:LOW:SEGM and CALC:LIM:UPP:SEGM or CALC:LIM:LOW:TRAC and CALC:LIM:UPP:TRAC to define a trace's current limits via the GPIB.

If a trace specifier is not used, the command defaults to trace A.

---

**CALCulate[1 | 2 | 3 | 4]:LIMit:FAIL?****query**

Returns the result of the last limit test; 0 for pass or 1 for fail.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :LIMit:FAIL?

Example Statements:            OUTPUT 711;"calculate2:lim:fail?"  
                                  OUTPUT 711;"Calc3:Lim:Fail?"

Return Format:                 Integer

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required:  no  
                                  Preset State:   +1  
                                  SCPI Compliance:  confirmed

**Description:**

This query returns "+0" if the trace passes the limit test. It returns a "+1" if the trace fails the limit test. Limit testing must be on (CALC:LIM:STAT ON) and a limit must be defined for the specified trace.

If limit testing is not on or limits are not defined, this query returns a +1 (fail).

Use the SYST:ERR? query to verify a failed limit test. If limit testing is not on, the SYST:ERR? query returns the message, "Limit testing is turned off." If limits are not defined the SYST:ERR? query returns the message, "Limits are undefined." If a valid limit test failed, the SYST:ERR? query does not return a message.

---

**Note**                    CALC:LIM:FAIL? returns +1 (fail) if limit testing is not turned on or limits are not defined.

---

---

**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:CLEar[:IMMediate]****command**

Deletes the lower limit line from the specified display.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:CLEar [:IMMediate]

Example Statements:        OUTPUT 711;":CALC4:LIMIT:LOW:CLE"  
                          OUTPUT 711;"calculate:lim:lower:cle:imm"

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   not applicable  
                          SCPI Compliance:   instrument-specific

Description:

To delete a lower limit line, send CALC:LIM:LOW:CLE. To delete an upper limit, send CALC:LIM:UPP:CLE.

You can delete part of a limit line if it consists of segments. See the CALC:LIM:LOW:SEGM:CLE command for information about deleting a segment of the lower limit line.

**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:MOVE:Y****command**

Moves all segments of the lower limit line up or down in the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:MOVE:Y <number> | <bound>

<number> ::= a real number (NRf data)  
                   limits: -9.9e37:9.9e37  
                   (depend upon current vertical/division unit)  
 <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"Calculate4:Lim:Lower:Move:Y 9.55849e+37"  
                               OUTPUT 711;"CALCULATE:LIM:LOWER:MOVE:Y -3.82565e+37"

Attribute Summary:        Option: not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

## Description:

This command increments or decrements all segments in a lower limit by the specified value along the Y-axis. The value is unitless and assumes the current vertical/division unit (returned with DISP:TRAC:Y:PDIV? UNIT).

To specify trace box A, send CALC1:LIM:LOW:MOVE:Y. To specify trace box B, send CALC2:LIM:LOW:MOVE:Y. To specify trace box C, send CALC3:LIM:LOW:MOVE:Y. To specify trace box D, send CALC4:LIM:LOW:MOVE:Y. If a trace specifier is not included in the command, the trace specifier defaults to trace box A.



**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:REPort[:DATA]?****query**

Returns the X-axis value of the failed points for the lower limit test.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:REPort [:DATA] ?

Example Statements:            OUTPUT 711;":calculate:lim:lower:rep?"  
                                   OUTPUT 711;"Calc:Limit:Low:Report:Data?"

Return Format:                   <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<X-axis value>,<X-axis value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<X-axis value>,<X-axis value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<X-axis value> ::= a real number (X-axis value of the failed point)
                   limits: -9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the X-axis value for data points which fail the lower limit test.

Data is not returned if limit testing is turned off (CALC:LIM:STAT OFF) or if all trace points are above the specified lower limit.

**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:REPort:YDATa?****query**

Returns the Y-axis value of the failed points for the lower limit test.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:REPort:YDATa?

Example Statements:            OUTPUT 711;"CALC4:LIMIT:LOW:REPORT:YDAT?"  
                                   OUTPUT 711;"calc3:limit:low:report:ydat?"

Return Format:                 <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<Y-axis value>,<Y-axis value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<Y-axis value>,<Y-axis value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<Y-axis value> ::= a real number
                  (Y-axis value of the failed point)
                  limits: - 9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the Y-axis value for data points which fail the lower limit test.

Data is not returned if limit testing is turned off (CALC:LIM:STAT OFF) or if all trace points are above the specified lower limit.

**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:SEGment****command/query**

Defines the lower limit as a series of line segments in the specified display.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:SEGment <BLOCK>

When data is ASCII-encoded, (FORMat:DATA ASCii), <BLOCK> takes the following form:

```
<LIMIT> ::= <segment>[,<segment> . . . .]
<segment> ::= <start_X-axis_value>, <start_Y-axis_value>
               <stop_X-axis_value>, <stop_Y-axis_value>
```

When data is binary-encoded, (FORMat:DATA REAL) <BLOCK> takes the following form:

```
<BLOCK> ::= #<byte>[<length_bytes>]<segment> [,<segment> . . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
            length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
<segment> ::= <start_X-axis_value>, <start_Y-axis_value>
               <stop_X-axis_value>, <stop_Y-axis_value>
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<start_X-axis_value> ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<start_Y-axis_value> ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<stop_X-axis_value>  ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<stop_Y-axis_value>  ::= a real number
                       limits: -9.9e+37 : 9.9e+37
```

Example Statements:       OUTPUT 711;" :CALCULATE2:LIM:LOW:SEGMENT 10, 2, 100, 3"  
                           OUTPUT 711;"calc:lim:low:segm 20000, -5, 3000, -5, 80000,  
                           -2, 90000,-2"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:SEGment?

Return Format:           definite length <BLOCK>

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not affected by Preset  
                           SCPI Compliance: instrument-specific

## CALCulate

### Description:

This command loads all segments of a limit. Each segment must consist of a start value (start\_X-axis\_value, start\_Y-axis\_value) and a stop value (stop\_X-axis\_value, stop\_Y-axis\_value).

The analyzer does not clear the previous lower limit definition when you send new segments. It only overwrites those portions of the limit redefined by the new segments. Send CALC:LIM:LOW:CLE to clear the previous limit.



---

**CALCulate[1 | 2 | 3 | 4]:LIMit:LOWer:TRACe[:IMMEDIATE]****command**

Converts the specified trace into a lower limit line.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:LOWer:TRACe [:IMMEDIATE]

Example Statements:        OUTPUT 711;"calc2:limit:low:trace:imm"  
                              OUTPUT 711;"Calc2:Limit:Low:Trace"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

CALCulate[1|2|3|4] specifies the trace box. Use CALC1 to specify trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D.

The limit line becomes active for the specified trace box.

**CALCulate[1 | 2 | 3 | 4]:LIMit:STATe****command/query**

Turns limit testing on and off for the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:STATe OFF | 0 | ON | 1

Example Statements:        OUTPUT 711; ":CALC2:LIM:STATE OFF"  
                               OUTPUT 711; "calc:limit:stat ON"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :LIMit:STATe?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: OFF (+0)  
                               SCPI Compliance: confirmed

**Description:**

When limit testing is on, the specified trace is evaluated against the limits defined in its upper and lower limit registers. If a trace specifier is not used, the command defaults to trace A. You can load these registers via the GPIB using the CALC:LIM:LOW:SEGM and the CALC:LIM:UPP:SEGM commands or the CALC:LIM:UPP:TRAC and CALC:LIM:LOW:TRAC commands.

To determine whether or not a trace is within the specified limits, you can send the CALC:LIM:FAIL query or monitor the bits in the Limit Fail condition register. (For more information, see "Limit Fail Register Set" in chapter 1.)

To return failed points on the X-axis, send the CALC:LIM:LOW:REP and the CALC:LIM:UPP:REP queries. To return failed points on the Y-axis, send the CALC:LIM:LOW:REP:YDAT and the CALC:LIM:UPP:REP:YDAT queries.

---

**Note**                    Limit lines are not automatically displayed when limit testing is enabled. To display limits you must send DISPLAY[:WINDow[1|2|3|4]]:LIMt:STATe ON.

---

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:CLEar[:IMMEDIATE]****command**

Deletes the upper limit line from the specified display.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:CLEar [:IMMEDIATE]

Example Statements:        OUTPUT 711;"Calc3:Limit:Upp:Clear:Imm"  
                          OUTPUT 711;"CALC:LIMIT:UPP:CLEAR"

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   not applicable  
                          SCPI Compliance:   instrument-specific

Description:

To delete an upper limit line, send CALC:LIM:UPP:CLE. To delete a lower limit line, send CALC:LIM:LOW:CLE.

You can delete part of a limit line if it consists of segments. See the CALC:LIM:UPP:SEGM:CLE command for information about deleting a segment of the upper limit line.



**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:MOVE:Y****command**

Moves all segments of the upper limit line up or down in the specified trace box.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:MOVE:Y <number> | <bound>

<number> ::= a real number (Nrf data)  
                   limits: -9.9e37:9.9e37  
                   (depend upon current Y-axis value)  
 <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;" :calc:lim:upper:move:y -3.86779e+37"  
                               OUTPUT 711;"Calc:Lim:Upper:Move:Y -6.89317e+37"

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

## Description:

This command increments or decrements all segments in a upper limit by the specified value along the Y-axis. The value is unitless and assumes the current vertical/division unit (returned with DISP:TRAC:Y:PDIV? UNIT).

To specify trace box A, send CALC1:LIM:UPP:MOVE:Y. To specify trace box B, send CALC2:LIM:UPP:MOVE:Y. To specify trace box C, send CALC3:LIM:UPP:MOVE:Y. To specify trace box D, send CALC4:LIM:UPP:MOVE:Y. If a trace specifier is not included in the command, the trace specifier defaults to trace box A.

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:REPort[:DATA]?****query**

Returns the X-axis value of the failed points for the upper limit test.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:REPort [:DATA] ?

Example Statements:            OUTPUT 711;"CALC4:LIM:UPPER:REP:DATA?"  
                                   OUTPUT 711;"calc3:lim:upper:rep?"

Return Format:                 <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<X-axis value>,<X-axis value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<X-axis value>,<X-axis value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<X-axis value> ::= a real number (X-axis value of the failed point)
                   limits: -9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the X-axis value for data points which fail the upper limit test.

Data is not returned if limit testing is turned off (CALC:LIM:STAT OFF) or if all trace points are below the specified upper limit.

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:REPort:YDATa?****query**

Returns the Y-axis value of the failed points for the upper limit test.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:REPort:YDATa?

Example Statements:            OUTPUT 711;":Calculate:Lim:Upp:Report:Ydat?"  
                                   OUTPUT 711;":CALCULATE:LIM:UPP:REPORT:YDAT?"

Return Format:                   <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<Y-axis value>,<Y-axis value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<Y-axis value>,<Y-axis value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<Y-axis value> ::= a real number (Y-axis value of the failed point)
                   limits: -9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the Y-axis value for data points which fail the upper limit test.

Data is not returned if limit testing is turned off (CALC:LIM:STAT OFF) or if all trace points are below the specified upper limit.

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:SEGment****command/query**

Defines the upper limit as a series of line segments in the specified display.

Command Syntax:           CALCulate:LIMit:UPPer:SEGment <BLOCK>

When data is ASCII-encoded, (FORMat:DATA ASCii) <BLOCK> take the following form:

```
<BLOCK> ::= <segment>[,<segment> . . .]
<segment> ::= <start_X-axis_value>, <start_Y-axis_value>
               <stop_X-axis_value>, <stop_Y-axis_value>
```

When data is binary-encoded, (FORMat:DATA REAL) <BLOCK> take the following form:

```
<BLOCK> ::= #<byte>[<length_bytes>]<segment> [<segment> . . .]
<byte> ::= one ASCII-encoded byte specifying the number
            of length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number
                   of data bytes to follow
<segment> ::= <start_X-axis_value>, <start_Y-axis_value>
               <stop_X-axis_value>, <stop_Y-axis_value>
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<start_X-axis_value> ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<start_Y-axis_value> ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<stop_X-axis_value>  ::= a real number
                       limits: -9.9e+37 : 9.9e+37
<stop_Y-axis_value>  ::= a real number
                       limits: -9.9e+37 : 9.9e+37
```

Example Statements:       OUTPUT 711;"calc2:lim:upper:segm 10, 2, 100, 3"  
                           OUTPUT 711;"Calc:Lim:Upp:Segment 20000, -5, 3000, -5,  
                           80000, -2, 90000,-2"

Query Syntax:            CALCulate[1|2|3|4]:LIMit:UPPer:SEGment?

Return Format:            definite length <BLOCK>

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not affected by Preset  
                           SCPI Compliance: instrument-specific

**Description:**

This command loads all segments of a limit. Each segment must consist of a start value (start\_X-axis\_value, start\_Y-axis\_value) and a stop value (stop\_X-axis\_value, stop\_Y-axis\_value).

The analyzer does not clear the previous upper limit definition when you send new segments. It only overwrites those portions of the limit redefined by the new segments. Send CALC:LIM:UPP:CLE to clear the previous limit.

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:SEGMent:CLEar****command**

Deletes a segment from the upper limit line.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:SEGMent:CLEar <number> | <bound>

<number> ::= a real number (NRf data)  
               limits: -9.9e37:9.9e37  
               (depend upon current range of the X-axis)

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"calculate4:lim:upp:segment:cle -3.21429e+37"  
                               OUTPUT 711;"Calculate4:Lim:Upp:Segment:Cle -2.22796e+37"

Attribute Summary:       Option: not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

## Description:

This command deletes any segment which contains the X-axis value you specify. Adjacent segments are not affected, although the limit line may be discontinuous.

The value entered for a limit line segment is unitless.

To delete all segments of an upper limit line, send CALC:LIM:UPP:CLE.

**CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:TRACe[:IMMediate]****command**

Converts the specified trace into an upper limit line.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :LIMit:UPPer:TRACe [:IMMediate]

Example Statements:        OUTPUT 711;":CALCULATE3:LIM:UPP:TRACE"  
                              OUTPUT 711;"calc:limit:upp:trac:immediate"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

CALCulate[1|2|3|4] specifies the trace box. Use CALC1 to specify trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D.

The limit line becomes active for the specified trace box.

**CALCulate[1 | 2 | 3 | 4]:MARKer:BAND:STARt****command/query**

Specifies the lowest frequency of the band in which power is calculated.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:BAND:STARt {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37  
                  <unit> ::= [HZ | S | ORD | COUNT | AVG | V | VPK | RPM | EU | REV]  
                  <step> ::= UP | DOWN  
                  <bound> ::= MAX | MIN

Example Statements:        OUTPUT 711;"calculate3:mark:band:star 0.0025 s"  
                              OUTPUT 711;"CALC:MARK:BAND:STAR 10000 HZ"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:BAND:STARt?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

This command defines the start value for the band used in calculating marker functions selected with the CALC:MARK:FUNC command. The specified value affects only the currently selected marker function.

The value specified with the CALC:MARK:BAND:STAR command must be less than the value specified with the CALC:MARK:BAND:STOP command.

---

**Note**                    If you want to move the frequency band below the current frequency band, you must reset the start value first. If you want to move the frequency band above the current frequency band, you must reset the stop value first.

---

To increment the value to the next largest point on the X-axis, send CALC:MARK:BAND:STAR UP.

To decrement the value to the next smallest point on the X-axis, send CALC:MARK:BAND:STAR DOWN.



You can also set the value with an expression. Send `CALC:MARK:BAND:STAR (CALC:MARK:X?)` to set the value to the current X-axis marker value.

If the X-axis is in time, this command specifies the start time for computation of time domain parameters. The default position is at the left edge of the trace.

The query returns the value of the current start frequency of the band (or the start time) in X-axis units. The value is returned even if the band markers are not on.

To determine the X-axis units, send `CALC:MARK:BAND:STAR? UNIT`.

**CALCulate[1 | 2 | 3 | 4]:MARKer:BAND:STOP****command/query**

Specifies the highest frequency of the band in which power is calculated.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:BAND:STOP {<number>[<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37  
  
<unit> ::= [HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV]  
  
<step> ::= UP|DOWN  
  
<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;" :CALC:MARK:BAND:STOP 75000 HZ"  
                              OUTPUT 711;"calculate2:mark:band:stop .0028 s"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:BAND:STOP?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

This command defines the stop value for the band used in calculating marker functions specified with the CALC:MARK:FUNC command. The specified value affects only the currently selected marker function.

The value specified with the CALC:MARK:BAND:STOP command must be greater than the value specified with the CALC:MARK:BAND:START command.

---

**Note**                    If you want to move the frequency band below the current frequency band, you must reset the start value first. If you want to move the frequency band above the current frequency band, you must reset the stop value first.

---

To increment the value to the next largest point on the X-axis, send CALC:MARK:BAND:STOP UP.

To decrement the value to the next smallest point on the X-axis, send CALC:MARK:BAND:STOP DOWN.

You can also set the value with an expression. Send  
CALC:MARK:BAND:STOP (CALC:MARK:X?) to set the value to the current X-axis marker value.

If the X-axis is in time, this command specifies the stop time for computation of time domain parameters.  
The default position is at the right edge of the trace.

The query returns the value of the current stop frequency of the band (or the stop time) in X-axis units.  
The value is returned even if the band markers are not on.

To determine the X-axis units, send CALC:MARK:BAND:STOP? UNIT.

**CALCulate[1 | 2 | 3 | 4]:MARKer:COUPled[:STATe]****command/query**

Couples the markers on all traces with the marker of the most active trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:COUPled[:STATe] OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;"Calc2:Marker:Coup:Stat ON"  
                               OUTPUT 711;"CALCULATE2:MARK:COUPLED ON"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:COUPled[:STATe] ?

Return Format:            Integer

Attribute Summary:       Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:   OFF (+0)  
                               SCPI Compliance:  instrument-specific

## Description:

This command moves the main maker of all traces to the same X-axis point as the marker of the most active trace. This ties the movement of the main markers together.

The most active trace is:

- Trace A if all traces are active or if trace A and trace B are the only active traces.
- Trace C if trace C and trace D are the only active traces.
- The active trace if only one trace is active.

The trace specifier is ignored.

The position of each marker is updated, even when the trace is not displayed. You cannot move a marker beyond the maximum number of points in the most active trace.

When coupled markers are used in a zoomed measurement (starting frequency > 0), the first point is assumed to be zero.

---

**Note**                    This command couples the markers for each trace by X-axis position; not X-axis values.

---

---

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABLE:CLEar[:IMMEDIATE]****command**

Clears all values in the specified data table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABLE:CLEar [:IMMEDIATE]

Example Statements:        OUTPUT 711;":calc:mark:dtable:cle"  
                              OUTPUT 711;"Calculate:Mark:Dtab:Clear:Imm"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

---

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABLE:COPIE[1 | 2 | 3 | 4]****command**

Copies the data table from one trace to another trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABLE:COPIE [1 | 2 | 3 | 4]

Example Statements:        OUTPUT 711; ":calc4:marker:dtab:copy1"  
                              OUTPUT 711; "CALC:MARK:DTABLE:COPIE3"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

Use the trace specifier, COPY[1|2|3|4], to indicate the source, which trace the data table is to be copied from. Use the trace specifier, CALCulate [1|2|3|4], to indicate the destination, which trace the data table is to be copied to.

1 specifies the data table appearing in trace box A, 2 specifies B, 3 specifies C, and 4 specifies trace box D. If a trace specifier is not used, the trace defaults to trace A.

---

**Note**                    The analyzer does not verify the traces are compatible.

---

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABle[:DATA]?****query**

Returns the dependent values in the specified data table.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle [:DATA] ?

Example Statements:            OUTPUT 711;"CALCULATE4:MARK:DTAB:DATA?"  
                                   OUTPUT 711;"calc4:marker:dtab?"

Return Format:                 <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<Y value>,<Y value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<Y value>, <Y value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<Y value> ::= a real number
           limits: -9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

For most types of measurement data and trace coordinates, all but Nyquist diagrams, polar diagrams, and orbit diagrams, the data table consists of three columns. The first column contains the label, the second column contains the independent value, and the third column contains the dependent value. This command returns the dependent values that appear in the third column. Use the command "CALC:MARK:Y? UNIT" to determine the unit.

For Nyquist diagrams, the data table consists of four columns. The first column contains the label, the second column contains the independent value (frequency), the third column contains the real value, and the fourth column contains the imaginary value. This command returns the real and imaginary values. Use the "CALC:MARK:X? UNIT" command to determine the unit associated with the real values. Use the "CALC:MARK:Y? UNIT" command to determine the unit associated with the imaginary values.

## CALCulate

For polar diagrams, the data table consists of four columns. The first column contains the label, the second column contains the independent value (frequency), the third column contains magnitude, and the fourth column contains phase. This command returns the magnitude and phase values. Use the “CALC:MARK:X? UNIT” command to determine the unit associated with the magnitude values. Use the “CALC:MARK:Y? UNIT” command to determine the unit associated with the phase values.

For orbit diagrams, the data table consists of four columns. The first column contains the label, the second column contains the independent value (time), the third column contains the X-axis value, and the fourth column contains the Y-axis value. This command returns the X-axis and Y-axis values. Use the “CALC:MARK:X? UNIT” command to determine the unit associated with the X-axis values. Use the “CALC:MARK:Y? UNIT” command to determine the unit associated with the Y-axis values.



**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABLE:X[:DATA]?****query**

Returns the X values in the specified data table.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :MARKer:DTABLE:X[:DATA] ?

Example Statements:            OUTPUT 711; ":Calc4:Marker:Dtab:X?"  
                                   OUTPUT 711; "CALC:MARK:DTABLE:X:DATA?"

Return Format:                   <DEF\_BLOCK>

When data is ASCII-encoded, (FORMat ASCii) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= [<X value>,<X value> . . .]
```

When data is binary-encoded, (FORMat REAL) <DEF\_BLOCK> takes the following form:

```
<DEF_BLOCK> ::= #<byte><length_bytes>[<X value>, <X value> . . .]
<byte> ::= one ASCII-encoded byte specifying the number of
           length bytes to follow
<length_bytes> ::= ASCII-encoded bytes specifying the number of
                   data bytes to follow
```

The following definitions apply to both ASCII- and binary-encoded data.

```
<X value> ::= a real number
           limits: -9.9e+37 : 9.9e+37
```

Attribute Summary:            Option: not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

Use the command "CALC:MARK:X? UNIT" to determine the unit.

---

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABle:X:DELeTe****command**

Deletes the selected entry in the data table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle:X:DELeTe

Example Statements:        OUTPUT 711;"calc:mark:dtable:x:delete"  
                              OUTPUT 711;"Calc3:Mark:Dtable:X:Delete"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

To select the X value entry in the data table, use the CALC:MARKER:DTAB:X:SELECT[:POINT] command.

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

To display the data table, use the DISPLAY[:WINDOW[1|2|3|4]]:DTABLE[:STATE] command.

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABLE:X:INSert****command/query**

Inserts an entry into the data table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABLE:X:INSert {<number>  
                              [<unit>] } |<bound>

<number> ::= a real number (NRf data)  
                              limits: -9.9e37:9.9e37

<unit> ::= [HZ | S | ORD | COUNT | AVG | V | VPK | RPM | EU | REV]

<bound> ::= MAX | MIN

Example Statements:        OUTPUT 711;":CALC3:MARK:DTABLE:X:INSERT 2.0516e+37"  
                              OUTPUT 711;"calc:mark:dtable:x:insert 6.26239e+36"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:DTABLE:X:INSert?

Return Format:             Real

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

**Description:**

If you want to insert entries in a specific order or if you are modifying an existing data table, use the CALC:MARKER:DTAB:X:SELECT[:POINT] command. The X value is inserted *before* the selected entry.

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

To display the data table, use the DISPLAY[:WINDow[1|2|3|4]]:DTABLE[:STATe] command.

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABle:X:LABel****command/query**

Loads a label for the selected entry in the data table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle:X:LABel '<STRING>'

<STRING> ::= ASCII characters - 0 through 255  
                  maximum number of characters: 16

Example Statements:        OUTPUT 711;"calc1:mark:dtab:x:lab 'Fundamental' "  
                              OUTPUT 711;"calc3:mark:dtable:x:label '                  Carrier' "

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle:X:LABel?

Return Format:             "<STRING>"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

To select the X value entry in the data table, use the CALC:MARKER:DTAB:X:SELECT[:POINT] command. The entry must exist in the data table before you assign a label to it.

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

To display the data table, use the DISPlay[:WINDow[1|2|3|4]]:DTABle[:STATe] command.

**CALCulate[1 | 2 | 3 | 4]:MARKer:DTABle:X:SElect[:POINT]****command/query**

Selects the data table entry.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle:X:SElect [:POINT] <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 1:50

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711; "Calc:Mark:Dtable:X:Select:Poin 23"  
                          OUTPUT 711; "CALC:MARKER:DTAB:X:SEL 32"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:DTABle:X:SElect [:POINT] ?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

This command selects the entry in the data table. Use it to direct the action of the following commands:

- CALC:MARK:DTAB:X:CHANGe
- CALC:MARK:DTAB:X:DELeTe
- CALC:MARK:DTAB:X:INSert
- CALC:MARK:DTAB:X:LABel

CALCulate[1|2|3|4] specifies the data table. Send CALC1 to specify the data table appearing in trace box A, CALC2 to specify B, CALC3 to specify C, and CALC4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

To display the data table, use the DISPLAY[:WINDow[1|2|3|4]]:DTABle[:STATe] command.

**CALCulate[1 | 2 | 3 | 4]:MARKer:FUNction****command/query**

Selects one of the analyzer's marker functions.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:FUNction OFF | 0 | HPOWer | THD |  
BPOWer | BRMS | SPOWer | OVERshoot | RTIME | STIME | DTIME | SSLevel |  
GMARgin | PMARgin | GCROSSover | PCROSSover | FREQuency | DAMPing |  
SINFo | WEIGHt | TPOWer | WPOWer

Example Statements:       OUTPUT 711;":calc:marker:func OVERSHOOT"  
OUTPUT 711;"Calculate:Mark:Func TPOWER"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:FUNction?

Return Format:            OFF | HPOW | THD | BPOW | BRMS | SPOW | OVER | RTIM | STIM | DTIM | SSL | GMAR  
| PMAR | GCR | PCR |   FREQ | DAMP | SINP | WEIG | TPOW | WPOW

Attribute Summary:       Option:   not applicable  
Synchronization Required: no  
Preset State:   not applicable  
SCPI Compliance: instrument-specific

## Description:

This command selects the marker function; send CALCulate:MARKer:FUNction:RESult? to read marker values.

To turn off the marker function, send CALC:MARK:FUNC OFF.

Marker functions depend on the type of measurement data. See table 6-4 for a listing of marker functions for each instrument mode. To define the range of the calculation, use the CALC:MARK:BAND commands.

The following marker functions are available for frequency data :

To select the harmonic power function, send CALC:MARK:FUNC HPOW.

To select the total harmonic distortion function, send CALC:MARK:FUNC THD.

To select the band power function, send CALC:MARK:FUNC BPOW.

To select the square root of the band power function, send CALC:MARK:FUNC BRMS.

To select the sideband power function, send CALC:MARK:FUNC SPOW.

## CALCulate

The following marker functions are available for frequency response data

(CALC:FEED 'XFR:POW:RAT [2,1|3,1|4,1|4,3]')

To select the gain margin power function, send CALC:MARK:FUNC GMAR.

To select the phase margin power function, send CALC:MARK:FUNC PMAR.

To select the gain crossover function, send CALC:MARK:FUNC GCR.

To select the phase crossover power function, send CALC:MARK:FUNC PCR.

To select the resonant frequency, send CALC:MARK:FUNC FREQ.

To select the damping function, send CALC:MARK:FUNC DAMP.

The following marker functions are available for time data (CALC:FEED 'XTIM:VOLT [1|2|3|4]')

To select the delay time function, send CALC:MARK:FUNC DTIM.

To select the maximum overshoot function, send CALC:MARK:FUNC OVER.

To select the rise time function, send CALC:MARK:FUNC RTIM.

To select the settling time function, send CALC:MARK:FUNC STIM.

To select the steady state level function, send CALC:MARK:FUNC SSL.

In correlation analysis instrument mode (INST:SEL CORR), the following marker functions are available:

To select the delay time function, send CALC:MARK:FUNC DTIM.

To select the maximum overshoot function, send CALC:MARK:FUNC OVER.

To select the rise time function, send CALC:MARK:FUNC RTIM.

To select the settling time function, send CALC:MARK:FUNC STIM.

To select the steady state level function, send CALC:MARK:FUNC SSL.

## CALCulate

In octave analysis instrument mode (INST:SEL OCT; Option 1D1) the following marker functions are available:

To select the band power function, send CALC:MARK:FUNC BPOW.

To select the square root of the band power function, send CALC:MARK:FUNC BRMS.

To select the overall band, send CALC:MARK:FUNC TPOW.

To select the weighted overall band, send CALC:MARK:FUNC WPOW.

In order analysis instrument mode (INST:SEL ORD; Option 1D0) the following marker functions are available:

To select the harmonic power function, send CALC:MARK:FUNC HPOW.

To select the total harmonic distortion function, send CALC:MARK:FUNC THD.

To select the band power function, send CALC:MARK:FUNC BPOW.

To select the square root of the band power function, send CALC:MARK:FUNC BRMS.

To select the sideband power function, send CALC:MARK:FUNC SPOW.

In histogram analysis instrument mode (INST:SEL HIST) the following marker functions are available for unfiltered time data (CALC:FEED 'XTIM:VOLT [1|2|3|4]'):

To select the delay time function, send CALC:MARK:FUNC DTIM.

To select the maximum overshoot function, send CALC:MARK:FUNC OVER.

To select the rise time function, send CALC:MARK:FUNC RTIM.

To select the settling time function, send CALC:MARK:FUNC STIM.

To select the steady state level function, send CALC:MARK:FUNC SSL.

---

**Note** Marker functions are not available for following types of trace data:  
Nyquist diagram (CALC:FORM NYQ)  
Polar diagram (CALC:FORM POL)  
Orbit diagram (CALC:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]')

---



Table 6-4. Marker Functions by Instrument Mode.

Marker Function	Instrument Mode								
	FFT			Swept Sine		Corr data	Octave data	Order data	Hist data
	Freq data	Freq resp	Time data	Freq data	Freq resp				
harmonic power CALC:MARK:FUNC HPOW	X			X				X	
total harmonic distortion CALC:MARK:FUNC THD	X			X				X	
band power CALC:MARK:FUNC BPOW	X	X	X	X	X	X	X	X	X
square root band power CALC:MARK:FUNC BRMS	X	X	X	X	X	X	X	X	X
sideband power CALC:MARK:FUNC SPOW	X			X				X	
gain margin CALC:MARK:FUNC GMAR		X			X				
phase margin CALC:MARK:FUNC PMAR		X			X				
gain crossover CALC:MARK:FUNC GCR		X			X				
phase crossover CALC:MARK:FUNC PCR		X			X				
resonant frequency CALC:MARK:FUNC FREQ		X			X				
damping CALC:MARK:FUNC DAMP		X			X				
delay time CALC:MARK:FUNC DTIM			X			X			X
maximum overshoot CALC:MARK:FUNC OVER			X			X			X
rise time CALC:MARK:FUNC RTIM			X			X			X
settling time CALC:MARK:FUNC STIM			X			X			X
steady state CALC:MARK:FUNC SSL			X			X			X
overall band CALC:MARK:FUNC TPOW							X		
weighted overall band CALC:MARK:FUNC WPOW							X		

## CALCulate

In addition to function calculations, use this command to determine if one of the filters (A-weight, B-weight, or C-weight) was applied to the measurement data or to determine the Z-axis value for the measurement data.

To determine if a filter was applied to the measurement data, send `CALC:MARK:FUNC WEIGHT`. Send the `CALC:MARK:FUNC:RES?` query to determine the results. The query returns 1 if the A-weight filter was applied to the measurement data; 2 if the B-weight filter was applied and 3 if the C-weight filter was applied. If a filter was not applied, the query returns 0 (false). The query returns 4 if the value is undefined. For example, if you are using two data registers, one which used the A-weight filter and the other which used the B-weight filter, the results are undefined.

To determine the Z-axis value for the measurement, send `CALC:MARK:FUNC SINFo`. Send the `CALC:MARK:FUNC:RES?` query to determine the Z-axis value for the measurement data. The Z-axis value indicates where the measurement data was extracted from a waterfall. It tells you when the measurement data was armed.

**CALCulate[1 | 2 | 3 | 4]:MARKer:FUNction:RESult?****query**

Returns the result of the calculation for the currently selected marker function.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :MARKer:FUNction:RESult?

Example Statements:            OUTPUT 711;"CALCULATE4:MARK:FUNCTION:RES?"  
                                   OUTPUT 711;"calc4:marker:func:result?"

Return Format:                 Real

Attribute Summary:            Option:   not applicable  
                                   Synchronization Required: no  
                                   Preset State:   not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This command returns the value of the marker function result calculation. To specify trace box A, send CALC1. To specify trace box B, send CALC2; trace box C, send CALC3; and trace box D, send CALC4. The trace specifier defaults to trace box A if the specifier is not used.

For example, send CALC3:MARK:FUNC GMAR then send CALC:MARK:FUNC:RES? to query the value of the gain margin of trace C. To determine the units, send CALC3:MARK:FUNC:RES? UNIT.

Refer to the CALC:MARK:FUNction command for a complete listing of the available marker functions.

---

**Caution**                    The analyzer returns the result 9.91E37, if it can not calculate a marker function.

---

In addition to function calculations, use this command to determine if one of the filters (A-weight, B-weight, or C-weight) was applied to the measurement data or to determine the Z-axis value for the measurement data.

## CALCulate

To determine if a filter was applied to the measurement data, send CALC:MARK:FUNC WEIGHT. Send the CALC:MARK:FUNC:RES? query to determine the results. The query returns 1 if the A-weight filter was applied to the measurement data; 2 if the B-weight filter was applied and 3 if the C-weight filter was applied. If a filter was not applied, the query returns 0 (false). The query returns 4 if the value is undefined. For example, if you are using two data registers, one which used the A-weight filter and the other which used the B-weight filter, the value is undefined.

To determine the Z-axis value for the measurement, send CALC:MARK:FUNC SINFo. Send the CALC:MARK:FUNC:RES? query to determine the Z-axis value for the measurement data. The Z-axis value indicates where the measurement data was extracted from a waterfall. It tells you when the measurement data was armed.

**CALCulate[1 | 2 | 3 | 4]:MARKer:HARMonic:COUNT****command/query**

Specifies the maximum number of harmonic markers for the display.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:HARMonic:COUNT <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:400

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711; "calc3:mark:harm:coun 5"  
                          OUTPUT 711; "CALC:MARK:HARMONIC:COUNT 3"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :MARKer:HARMonic:COUNT?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

The value can be specified numerically or with nonnumeric parameters. The actual number of harmonics is determined by the analyzer.

To increase the number of harmonic markers by one, send CALC:MARK:HARM:COUN UP.

To decrease the number of harmonic markers by one, send CALC:MARK:HARM:COUN DOWN.

The query returns the number of harmonic currently specified for the display. The value is returned even if the harmonic markers are not on.

**CALCulate[1 | 2 | 3 | 4]:MARKer:HARMonic:FUNDamental****command/query**

Specifies the fundamental frequency for harmonic markers and calculations.

Command Syntax:            CALCulate [1 | 2 | 3 | 4] :MARKer:HARMonic:FUNDamental {<number> [ <unit> ] } | <step> | <bound>

<number> ::= a real number (NRF data)  
                  limits: -9.9e37:9.9e37

<unit> ::= [HZ | ORD | CPM]

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:        OUTPUT 711; "calc:mark:harm:fund 440 hz"  
                              OUTPUT 711; "CALCULATE:MARKER:HARM:FUND 1750 RPM"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:HARMonic:FUNDamental?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The value can be specified numerically or with nonnumeric parameters.

To increment the value to the next largest point on the X-axis, send CALC:MARK:HARM:FUND UP.

To decrement the value to the next smallest point on the X-axis, send CALC:MARK:HARM:FUND DOWN.

You can also set the value with an expression. Send CALC:MARK:FUND (CALC:MARK:X?) to set the value to the current X-axis marker value.

The query returns the value of the fundamental frequency (in X-axis units) currently used for harmonic markers and calculations. The value is returned even if the harmonic markers are not on.

To determine the X-axis units, send CALC:MARK:HARM:FUND? UNIT.

**CALCulate[1 | 2 | 3 | 4]:MARKer:MAXimum[:GLOBal]****command**

Moves the main marker to the highest point on the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:MAXimum[:GLOBal]

Example Statements:        OUTPUT 711;" :Calc:Mark:Maximum"  
                              OUTPUT 711;"CALC:MARKER:MAX:GLOB"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

This command moves the marker to the highest peak one time. Another command —CALC:MARK:MAX:TRAC—controls a marker function that automatically moves the marker to the highest peak each time the trace is updated.

The specified trace does not need to be displayed for the marker to move to the highest peak.

**CALCulate[1 | 2 | 3 | 4]:MARKer:MAXimum[:GLOBal]:TRACk****command/query**

Turns the peak tracking function on or off.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:MAXimum[:GLOBal] :TRACk OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;"calculate2:mark:maximum:glob:trac OFF"  
                              OUTPUT 711;"Calculate3:Mark:Maximum:Trac ON"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:MAXimum[:GLOBal] :TRACk?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  OFF (+0)  
                              SCPI Compliance:  instrument-specific

**Description:**

When peak tracking is enabled, the analyzer automatically positions the main marker on the largest peak of the specified trace each time the trace is updated.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D. The trace does not need to be displayed nor must it be the primary active trace.

To move the marker to the highest peak one time, use the CALC:MARK:MAX:GLOB command.



**CALCulate[1 | 2 | 3 | 4]:MARKer:MAXimum:LEFT****command**

Moves the main marker one peak to the left of its current location on the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:MAXimum:LEFT

Example Statements:        OUTPUT 711;":CALC3:MARKER:MAX:LEFT"  
                              OUTPUT 711;"calc:mark:maximum:left"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

**Description:**

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

A peak is a local maximum on the displayed trace. The slope of a trace is positive to the left of a peak and negative to the right. In addition, the slope on one side of a peak must not change for at least one vertical division (one-third division on both sides if in octave analysis instrument mode or one-tenth division (one decade) if using logarithmic scaling of the Y-axis).

This command only finds peaks that are at least one point to the left of the current marker position. If the analyzer does not find a peak, the marker does not move. You can increase the number of peaks found by the analyzer by decreasing the value of vertical scale division (DISP:TRAC:Y:PDIV).

**CALCulate[1 | 2 | 3 | 4]:MARKer:MAXimum:RIGHT****command**

Moves the main marker one peak to the right of its current location on the specified trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:MAXimum:RIGHT

Example Statements:        OUTPUT 711; "Calculate4:Mark:Max:Right"  
                              OUTPUT 711; "CALC4:MARKER:MAX:RIGH"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

A peak is a local maximum on the displayed trace. The slope of a trace is positive to the left of a peak and negative to the right. In addition, the slope on one side of a peak must not change for at least one vertical division (one-third division on both sides if in octave analysis instrument mode or one-tenth division (one decade) if using logarithmic scaling of the Y-axis).

This command only finds peaks that are at least one point to the right of the current marker position. If the analyzer does not find a peak, the marker does not move. You can increase the number of peaks found by the analyzer by decreasing the value of vertical scale division (DISP:TRAC:Y:PDIV).

**CALCulate[1 | 2 | 3 | 4]:MARKer:MODE****command/query**

Selects absolute or relative marker values.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:MODE ABSolute|RELative

Example Statements:        OUTPUT 711;" :calculate4:mark:mode ABSOLUTE"  
                              OUTPUT 711;"Calc:Mark:Mode RELATIVE"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:MODE?

Return Format:             ABS|REL

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  ABSolute  
                              SCPI Compliance:  instrument-specific

**Description:**

To select relative marker values, send CALC:MARK:MODE REL. A marker reference is displayed and marker values are reported as distances between the reference point and the relative marker position.

To select absolute marker values, send CALC:MARK:MODE ABS. The marker values are reported as the position of the marker on the trace.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.





**Note** You cannot move the main marker beyond the maximum displayed point nor below the minimum displayed point.

Table 6-5. Number of displayed data points in variable resolution  
FFT Instrument Mode

Resolution	Baseband		Zoom (Start frequency $\neq$ 0)	
	Number of frequency points (frequency data)	Number of real time points (time data)	Number of frequency points (frequency data)	Number of complex time points (real/ imaginary pairs) (time data)
100	101	256	101	128
200	201	512	201	256
400	401	1024	401	512
800	801	2048	801	1024

Correlation Instrument Mode  
(no complex data)

Resolution	Auto- and cross correlation (real data)			Time domain (real data)
	0 to T/2	-T/2 to T/2	-T/4 to T/4	
100	128	256	128	256
200	256	512	256	512
400	512	1024	512	1024
800	1024	2048	1024	2048

Table 6-5. Number of displayed data points in variable resolution (continued)

**Swept Sine Instrument Mode  
(FREQ:RES:AUTO OFF)**

FREQ:RES (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to  $FREQ:SPAN / FREQ:RES (PNT/SWP)$ . With PCT, the spacing between measurement points is a percentage of the total frequency span.

**Swept Sine Instrument Mode  
(FREQ:RES:AUTO ON)**

FREQ:RES:AUTO:MIN (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to  $FREQ:SPAN / FREQ:RES:AUTO:MIN$ . With PCT, the spacing between measurement points is a percentage of the total frequency span.

**CALCulate[1 | 2 | 3 | 4]:MARKer:REFerence:X****command/query**

Specifies the marker reference X-axis position.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:REFerence:X {<number> [<unit>]} |  
                              <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37

<unit> ::= [HZ | S | ORD | COUNT | AVG | V | VPK | RPM | EU | REV]

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:        OUTPUT 711; " :Calc4:Marker:Ref:X 1.8207e+37"  
                              OUTPUT 711; "CALC:MARK:REFERENCE:X -5.84172e+37"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:REFerence:X?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +0.0  
                              SCPI Compliance: instrument-specific

Description:

This command specifies the absolute X-axis position for the reference marker.

To specify the marker reference X-axis position relative to the main marker's position, use the CALC:MARK:X:REL command.







**CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:COUNT****command/query**

Specifies the number of sideband markers for the display.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:SIDeband:COUNT <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:200

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711; ":calc4:marker:sid:count 2"  
                          OUTPUT 711; "CALC:MARK:SID:COUN 6"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer:SIDeband:COUNT?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

The value can be specified numerically or with nonnumeric parameters.

To increase the number of sideband markers by one, send CALC:MARK:HARM:COUN UP.

To decrease the number of sideband markers by one, send CALC:MARK:HARM:COUN DOWN.

The query (CALC:MARK:HARM:COUNT?) returns the number of sideband markers currently specified for the display. The value is returned even if the sideband markers are not on.



**CALCulate[1 | 2 | 3 | 4]:MARKer[:STATe]****command/query**

Turns on the main markers or turns off all markers and marker functions for a selected trace.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer [:STATe] OFF | 0 | ON | 1

Example Statements:        OUTPUT 711; ":CALCULATE3:MARK OFF"  
                               OUTPUT 711; "calculate:mark:stat OFF"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MARKer [:STATe] ?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: ON (+1)  
                               SCPI Compliance: instrument-specific

**Description:**

To display the main marker and its annotation, send CALC:MARK ON. The analyzer displays the X-axis and Y-axis values at the top of the grid.

To disable the display of the main markers and the marker reference for the active trace, send CALC:MARK OFF.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

---

**Note**                   All markers and marker functions operate on the specified trace, whether or not the trace is displayed.

---

**CALCulate[1 | 2 | 3 | 4]:MARKer:X[:ABSolute]****command/query**

Specifies the main marker's X-axis position.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:X[:ABSolute] {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [HZ | S | ORD | COUNT | AVG | V | VPK | RPM | EU | REV]

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711;"Calculate3:Mark:X:Abs 7.75161e+37"  
                          OUTPUT 711;"CALC4:MARKER:X -3.56784e+35"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :MARKer:X[:ABSolute] ?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.0  
                          SCPI Compliance: instrument-specific

## Description:

This command specifies the main marker's X-axis position. Send CALC:MARK:X? UNIT to determine the units for the X-axis.

To specify the main marker's X-axis position by display point number, use the CALC:MARK:POSITION:POINT command.

To specify the main marker's independent axis position for orbits, Nyquist diagrams and polar diagrams, send the CALC:MARK:POSITION command.

---

**Note**                    You can not move the main marker beyond the maximum displayed X-axis value nor below the minimum displayed X-axis value.

---

In octave analysis instrument mode (INST:SEL OCT), CALC:MARK:X MAX moves the main marker to the far right band.



**CALCulate[1 | 2 | 3 | 4]:MARKer:Y[:ABSolute]?****query**

Returns the main marker's Y-axis position.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :MARKer:Y[:ABSolute] ?

Example Statements:            OUTPUT 711;"CALCULATE4:MARK:Y:ABS?"  
                                  OUTPUT 711;"calc3:marker:y?"

Return Format:                 Real

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: instrument-specific

Description:

This query always returns the Y-axis position of the main marker, even if the marker is not currently displayed on the analyzer's screen. The returned value tells you the amplitude of the specified trace at the marker's X-axis position (specified with CALC:MARK:X or CALC:MARK:POS:POIN).

Send CALC:MARK:Y? UNIT to determine the units for the Y-axis.



**CALCulate[1 | 2 | 3 | 4]:MARKer:Y:RELative****command/query**

Specifies the marker reference Y-axis position relative to the main marker.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MARKer:Y:RELative <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: -150:150

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711; ":Calculate2:Mark:Y:Relative 109.477"  
                          OUTPUT 711; "CALC:MARKER:Y:REL 52.6622"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :MARKer:Y:RELative?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.0  
                          SCPI Compliance: instrument-specific

**Description:**

This command specifies the marker reference Y-axis position relative to the main marker's position.

To specify an absolute Y-axis position for the marker reference, use the CALC:MARK:REF:Y command.



**CALCulate[1 | 2 | 3 | 4]:MATH:DATA****command/query**

Loads a complete set of math definitions.

Command Syntax:           CALCulate:MATH:DATA <USER>

```

<USER> ::= <file_type><function_1><function_2> <function_3>
          <function_4><function_5><constant_1><constant_2>
          <constant_3><constant_4><constant_5>

<file_type> ::= 1503 specifies math table
<function_1> ::= 270 bytes specifying function expression,
                 terminating with null character
<function_2> ::= 270 bytes specifying function expression,
                 terminating with null character
<function_3> ::= 270 bytes specifying function expression,
                 terminating with null character
<function_4> ::= 270 bytes specifying function expression,
                 terminating with null character
<function_5> ::= 270 bytes specifying function expression,
                 terminating with null character

```

270 bytes must be sent when specifying a function expression. All characters following the first null character are discarded.

```

<constant_1> ::= <real_part_constant><imaginary_part_constant>
<constant_2> ::= <real_part_constant><imaginary_part_constant>
<constant_3> ::= <real_part_constant><imaginary_part_constant>
<constant_4> ::= <real_part_constant><imaginary_part_constant>
<constant_5> ::= <real_part_constant><imaginary_part_constant>
<real_part_constant> ::= 8 byte floating point number
                       (REAL variable in Instrument BASIC)
<imaginary_part_constant> ::= floating point number           8 byte
                              (REAL variable in Instrument BASIC)

```

Example Statements:       OUTPUT 711;"CALC:MATH:DATA #414321503 . . .  
see example programs LOADMATH and GETMATH

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :MATH:DATA?

Return Format:            definite length <USER>

Attribute Summary:       Option: not applicable  
Synchronization Required: no  
Preset State: not affected by Preset  
SCPI Compliance: instrument-specific

## CALCulate

### Description:

This command allows you to transfer a complete set of math definitions—the same information contained in a math file—between the analyzer and your controller.

When you transfer a set of math definitions to the analyzer, you can use either the definite or indefinite length block syntax. When the analyzer returns the set of math definitions, it always uses the definite length block syntax. See “Block Parameters” in the GPIB Programmer’s Guide for more information.

The MMEM:STOR:MATH command and the MMEM:LOAD:MATH command also transfer a complete set of math definitions using one of the analyzer’s mass storage devices. See these commands for more information about loading and storing data in the function registers and in the constant registers.

**CALCulate[1 | 2 | 3 | 4]:MATH[:EXPRession[1 | 2 | 3 | 4 | 5]]****command/query**

Defines a math function.

Command Syntax:           CALCulate:MATH[:EXPRession[1|2|3|4|5]] <EXPR>

                  <EXPR> ::= ([<expr\_element>]...)

                  <expr\_element> ::= see operations and operands listed below

Example Statements:        OUTPUT 711;"Calculate:Math:Expr2 (K1\*FRES) "  
                              OUTPUT 711;"CALC:MATH (TIME1-TIME2) "

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MATH[:EXPRession[1 | 2 | 3 | 4 | 5]] ?

Return Format:             STRING

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

Description:

This command loads an expression into one of five math function registers. The register defaults to F1 if you do not specify the register with :EXPR[1|2|3|4|5].

Before you can display the results of a trace math function, you must load the function definition into one of the analyzer's five function registers: F1 through F5. Once you have loaded the function register with CALC:MATH:EXPR, you execute the expression and display the results with the CALC:MATH:SEL command. CALC:MATH:STAT must be ON.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

To define trace math functions, combine the elements (listed below) according to the rules of standard algebraic notation. Use parentheses to control the order of operations.

- Operations
  - AWEIGHT Apply A-weight filter
  - BWEIGHT Apply B-weight filter
  - CWEIGHT Apply C-weight filter
  - CONJ Complex Conjugate
  - DIFF Differentiate
  - DJOM Divide by  $j \omega$
  - EXP Exponential
  - FFT Fast Fourier Transform
  - IFFT Inverse Fast Fourier Transform
  - INTEG Integrate
  - IMAG Imaginary Part
  - LN Natural Logarithm

## CALCulate

- MAG Magnitude
- PSD Power Spectral Density
- REAL Real Part
- SQRT Square Root
- XJOM multiply by  $j \omega$
- + Add
- - Subtract
- \* Multiply
- / Divide
- Operands
  - D1|D2|D3|D4|D5|D6|D7|D8 Contents of data registers
  - F1|F2|F3|F4|F5 Contents of function registers
  - K1|K2|K3|K4|K5 Contents of constant registers
  - Measurement Data (depends on instrument mode)  
[1|2|3|4] specifies which trace contains the measurement data. Channels 3 and 4 are only available with Option AY6.
    - ACORR[1|2|3|4] Autocorrelation (INST:SEL CORR only)
    - CDF[1|2|3|4] Cumulative Density Function (INST:SEL HIST only)
    - COH[21|31|41|43] Coherence
    - CPOW[1|2|3|4] Composite Power (INST:SEL ORD only; Option 1D0)
    - CSPEC[21|31|41|43] Cross Spectrum
    - FRES[21|31|41|43] Frequency Response
    - HIST[1|2|3|4] Histogram (INST:SEL HIST only)
    - LSPEC[1|2|3|4] Linear Spectrum
    - NVAR[1|2|3|4] Normalized Variance (INST:SEL SINE only; Option 1D2)
    - PDF[1|2|3|4] Probability Density Function (INST:SEL HIST only)
    - PSPEC[1|2|3|4] Power Spectrum
    - RPM RPM Profile (INST:SEL ORD only; Option 1D0)
    - TIME[1|2|3|4] Time Data
    - TIME[1|2|3|4] Resampled Time (INST:SEL ORD only; Option 1D0)
    - TIME[1|2|3|4] Unfiltered Time (INST:SEL HIST only)
    - TRACK[1|2|3|4|5][1|2|3|4] Order Track (INST:SEL ORD only; Option 1D0)  
[1|2|3|4|5] specifies which order track
    - WTIME[1|2|3|4] Windowed Time Data
    - XCORR[21|31|41|43] Cross Correlation (INST:SEL CORR only)

Refer to online help or the Agilent 35670A Operator's Guide for more information on math operations.

**CALCulate[1 | 2 | 3 | 4]:MATH:SElect****command/query**

Selects and displays the designated math function, if CALC:MATH:STAT ON.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MATH:SElect { F1 | F2 | F3 | F4 | F5 }

Example Statements:       OUTPUT 711;":CALC2:MATH:SEL:F5"  
                               OUTPUT 711;calc:math:select F1"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MATH:SElect?

Return Format:             F1 | F2 | F3 | F4 | F5

Attribute Summary:       Option:   not applicable  
                               Synchronization Required:   no  
                               Preset State:   F1  
                               SCPI Compliance:   instrument-specific

**Description:**

The results are displayed in the specified trace box. Omit the specifier or send CALC1 for trace A, CALC2 for trace B, CALC3 for trace C, or CALC4 for trace D.

Load the math function into the specified function register with the CALC:MATH:EXPR command. An error is generated if an expression contains operands not available in the selected instrument mode.

To display a math constant as a trace, create a math function with the CALC:MATH:EXPR command. For example, CALC:MATH:EXPR1 K1 loads the value of the math constant register, K1, into the math function register, F1. Then use this command (CALC1:MATH:SEL F1) to display the math constant (K1) in trace box A.

---

**Note**                    CALC[1|2|3|4]:MATH:STATe must be ON for the specified trace box.

---

**CALCulate[1 | 2 | 3 | 4]:MATH:STATe****command/query**

Evaluates the currently selected math operation for the specified trace and displays the results.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :MATH:STATe OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;"calculate2:math:state OFF"  
OUTPUT 711;"Calc:Math:State ON"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :MATH:STATe?

Return Format:             Integer

Attribute Summary:       Option:   not applicable  
Synchronization Required: no  
Preset State:   OFF (+0)  
SCPI Compliance: confirmed

**Description:**

CALC:MATH:STATE must be ON to perform math operations.

Use CALCulate [1|2|3|4] to specify the trace. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

To define a math function, use CALC:MATH:EXPR or CALC:MATH:DATA. (To define the value of the constant registers, use the CALC:MATH:CONS command.) Select the function with the CALC:MATH:SEL command. Execute the function and display the results with the CALC:MATH:STAT ON command.

CALC:MATH:STATE OFF turns off math operations. You cannot execute or display a math operation unless CALC:MATH:STATE is ON.

---

**Note**                   Each trace box always has a selected math function. When the analyzer receives the CALC[1|2|3|4]:MATH:STAT ON command, it evaluates the selected function and displays the results in the specified trace box. If a trace box is not specified, the function defaults to trace box A.

When the analyzer receives the CALC[1|2|3|4]:MATH:STAT off, it stops evaluating the function. The analyzer returns the data it was displaying in the trace box before receiving the CALC:MATH:STAT ON command.

---



---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis:COpy****command**

Copies the contents of the curve fit table into the synthesis table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :SYNThesis:COpy CFIT

Example Statements:       OUTPUT 711;" :CALC:SYNTHESIS:COpy CFIT"  
                          OUTPUT 711;"calc:synthesis:copy CFIT"

Attribute Summary:       Option:   1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

This command overwrites the synthesis table with the contents of the curve fit table. You cannot recover the contents of the previous synthesis table after sending this command.

The analyzer does not copy "engineering units." Use the CALC:SYNT:GAIN command to simulate engineering units with synthesis.

---

**Note**                   This command is not trace specific. It ignores the trace specifier.

---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis:DATA****command/query**

Loads values into the synthesis table.

Command Syntax:           CALCulate:SYNThesis:DATA <BLOCK>

                  <BLOCK> ::= see Description

Example Statements:        OUTPUT 711;"Calculate3:Synt:Data BLOCK"  
                          OUTPUT 711;"CALCULATE2:SYNT:DATA BLOCK"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :SYNThesis:DATA?

Return Format:             definite length <BLOCK>

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

This command transfers a complete synthesis table from your controller to the analyzer.

When you transfer a synthesis table to the analyzer, you must use the definite length block syntax. Data must be 64-bit binary floating-point numbers (see the FORMat[:DATA] REAL command). The elements of the definite length block for a synthesis table are defined below.

The first point specifies the table type; followed by three points which specify the size of the table. The next 80 points are the complex pairs. The remaining points are descriptors. Points 152 - 174 are used in curve fit tables. Points 173 - 175 are adjusters; gain, frequency scale and time delay.

See "Block Data" in the GPIB Programmer's Guide for more information about transferring block data.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

```

<BLOCK> ::= #41400<Point1>Point2> . . . <Point175>
<Point1> ::= Table type
           <0> = pole zero
           <1> = pole residue
           <2> = polynomial
<Point2> ::= number_of_lines_in_left_column
<Point3> ::= number_of_lines_in_right_column
<Point4> ::= number_of_lines_in_Laurent_column
<Point5> ::= real_part_first_term_in_left_column
<Point6> ::= imaginary_part_first_term_in_left_column
<Point7> ::= real_part_second_term_in_left_column
<Point8> ::= imaginary_part_second_term_in_left_column

```

```

      .
      .
      .
<Point47> ::= real_part_first_term_in_right_column
<Point48> ::= imaginary_part_first_term_in_right_column
      .
      .
      .
<Point89> ::= real_part_first_term_in_Laurent_column
<Point90> ::= imaginary_part_first_term_in_Laurent_column
      .
      .
      .
<Point131> ::= first_curve_fit_term_left_column
<Point132> ::= second_curve_fit_term_left_column
<curve_fit_term> ::= 0
      .
      .
      .
<Point152> ::= first_curve_fit_term_right_column
<Point153> ::= second_curve_fit_term_right_column
<curve_fit_term> ::= 0
      .
      .
      .
<Point173> ::= gain
<Point174> ::= frequency_scale
<Point 175> ::= time_delay

```

---

**Note**            If a curve fit term = 1, “fxd” (fixed) appears by the curve fit term in the table. It has no effect on synthesis.

---

---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis:DESTination****command/query**

Selects the data register for the results of the synthesis operation.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :SYNThesis:DESTination  
                              D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Example Statements:        OUTPUT 711;" :calc3:synt:destination D2"  
                              OUTPUT 711;"Calc:Synthesis:Dest D3"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :SYNThesis:DESTination?

Return Format:             D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: D8  
                              SCPI Compliance: instrument-specific

Description:

This command specifies which data register holds the synthesis of the intermediate and final synthesis models. The default register is D8.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis:FSCale****command/query**

Specifies a frequency scale for the synthesis operation.

Command Syntax:        CALCulate [1 | 2 | 3 | 4] :SYNThesis:FSCale <number> | <step> |  
                             <bound>

<number> ::= a real number (NRf data)  
                             limits: 1e-6:1e6

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"CALC2:SYNTHESIS:FSC 300318"  
                             OUTPUT 711;"calculate3:synt:fsc 669766"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :SYNThesis:FSCale?

Return Format:            Real

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                             Synchronization Required: no  
                             Preset State: +1.0  
                             SCPI Compliance: instrument-specific

**Description:**

This command scales the synthesis model along the X-axis by  $f$ /frequency scale, where  $f$  is frequency in Hz. The frequency scale must be a positive value.

The value can be used to scale poles and zeros from Hz to radians-per-second by setting the scaling value to  $1 / (2 \pi)$ .

Pole and zero terms are not multiplied by  $1$ /frequency scale.

---

**Note**

This command must be sent before CALC:SYNT.

This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:SYNTHesis:GAIN****command/query**

Specifies the gain constant, K, for a synthesis operation.

Command Syntax:           CALCulate:SYNTHesis:GAIN <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37 (excluding 0.0)  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;" :Calculate3:Synt:Gain 4.33554e+37"  
                          OUTPUT 711;"CALC:SYNT:GAIN 1.3059e+37"

Query Syntax:             CALCulate[1|2|3|4]:SYNTHesis:GAIN?

Return Format:            Real

Attribute Summary:       Option: 1D3 Curve Fit/Synthesis  
                          Synchronization Required: no  
                          Preset State: +1.0  
                          SCPI Compliance: instrument-specific

Description:

This command specifies the desired gain of a synthesized frequency response function.

The gain constant, K, is unitless.

---

**Note**

This command is not trace specific. It ignores the trace specifier.

The limits exclude 0.0.

---

---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis[:IMMEDIATE]****command**

Creates a frequency response curve from the synthesis table.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :SYNThesis [:IMMEDIATE]

Example Statements:        OUTPUT 711;"calc2:synthesis:imm"  
                              OUTPUT 711;"Calc:Synthesis"

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

This command creates a frequency response curve based on the current synthesis table.

Values for the table are entered with the CALC:SYNT:DATA command. The results of the synthesis operation are stored in the synthesis data register specified by the CALC:SYNT:DEST command.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

---

**CALCulate[1 | 2 | 3 | 4]:SYNThesis:SPACing****command/query**

Specifies a linear or logarithmic scale for the X-axis data spacing.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :SYNThesis:SPACing LINear|LOGarithmic

Example Statements:        OUTPUT 711;" :CALC3:SYNTHESIS:SPAC LOGARITHMIC"  
                              OUTPUT 711;"calc:synthesis:spac LINEAR"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :SYNThesis:SPACing?

Return Format:            LIN|LOG

Attribute Summary:        Option:   1D3 Curve Fit/Synthesis  
                              Synchronization Required: no  
                              Preset State: LIN  
                              SCPI Compliance: instrument-specific

Description:

To specify a linear spacing of points along the X-axis, send CALC:SYNT:SPAC LIN.

To specify a logarithmic scale, send CALC:SYNT:SPAC LOG.

This command should not be confused with the DISPLAY:TRACE:X:SPACing command which changes the X-axis display grid between linear and logarithmic spacing.

---

**Note**

This command is not trace specific. It ignores the trace specifier.

---





**CALCulate[1 | 2 | 3 | 4]:SYNThesis:TTYPe****command/query**

Converts the synthesis table to another table format.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :SYNThesis:TTYPe PZERo | PFRaction | POLY-nomial

Example Statements:        OUTPUT 711; ":calc2:synt:ttype PZERO"  
                               OUTPUT 711; "Calc:Synthesis:Ttyp POLYNOMIAL"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :SYNThesis:TTYPe?

Return Format:             PZER | PFR | POLY

Attribute Summary:        Option: 1D3 Curve Fit/Synthesis  
                               Synchronization Required: no  
                               Preset State: PZER  
                               SCPI Compliance: instrument-specific

Description:

To convert the synthesis table to pole-zero format, send CALC:SYNT:TTYP PZER.

To convert the synthesis table to partial-fraction format, send CALC:SYNT:TTYP PFR. This format is identified as pole-residue in the table.

To convert the synthesis table to polynomial format, send CALC:SYNT:TTYP POLY.

The analyzer ignores this command if the table already exists in the specified format.

**Note**

Table conversions between formats are not allowed if the table data represents a non-Hermitian symmetric system. Hermitian symmetry is most easily defined in the polynomial table format: all numerator and denominator coefficients must be real.

This command is not trace specific. It ignores the trace specifier.

**CALCulate[1 | 2 | 3 | 4]:UNIT:AMPLitude****command/query**

Selects the unit of amplitude for the Y-axis scale.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:AMPLitude PEAK | PP | RMS

Example Statements:        OUTPUT 711;"CALC3:UNIT:AMPL PP"  
                               OUTPUT 711;"calculate4:unit:ampl PP"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:AMPLitude?

Return Format:             PEAK | PP | RMS

Attribute Summary:        Option:   not applicable  
                               Synchronization Required:   yes  
                               Preset State:   RMS (Channel 1 and Channel 3)  
   PEAK (Channel 2 and Channel 4)  
                               SCPI Compliance:   instrument-specific

Description:

To display peak amplitude, send CALC:UNIT:AMPL PEAK.

To display peak-to-peak amplitude, send CALC:UNIT:AMPL PP.

To display RMS amplitude, send CALC:UNIT:AMPL RMS.

The default value is dependent upon the selected measurement data (CALC:FEED). Table 6-6 indicates valid unit selections for the CALC:UNIT:AMPL command. If measurement data does not appear in the table, you are not permitted to select the amplitude. In this case, a query returns a null string. See “Determining Units” in Appendix E for information about available Y-axis units.

Table 6-6. Valid Unit Selections for CALC:UNIT:AMPL

Measurement Data CALC:FEED command	CALC:UNIT:AMPL		
	RMS	PEAK	PP
Auto correlation CALC:FEED 'XTIM:CORR'		X	
Capture buffer CALC:FEED 'TCAP'		X	
Coherence CALC:FEED 'XFR:POW:COH'			
Composite Power CALC:FEED 'XFR:POW:COMP'	X	X	X
Cross Correlation CALC:FEED 'XTIM:CORR:CROS'		X	
Cross Spectrum CALC:FEED 'XFR:POW:CROS'	X	X	X
Cumulative Density Function CALC:FEED 'XTIM:VOLT:CDF'			
Frequency Response CALC:FEED 'XFR:POW:RAT'			
Histogram CALC:FEED 'XTIM:VOLT:HIST'			
Linear Spectrum CALC:FEED 'XFR:POW:LIN'	X	X	X
Order Track CALC:FEED 'XORD:TRACK'	X	X	X
Power Spectrum CALC:FEED 'XFR:POW'	X	X	X
Probability Density Function CALC:FEED 'XTIM:VOLT:PDF'			
RPM Profile CALC:FEED 'XRPM:PROF'			
Time CALC:FEED 'XTIM:VOLT'		X	
Windowed Time CALC:FEED 'XTIM:VOLT:WIND'		X	

**CALCulate[1 | 2 | 3 | 4]:UNIT:ANGLE****command/query**

Specifies the unit for phase coordinates.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:ANGLE DEGR|RADian

Example Statements:        OUTPUT 711;"CALC:UNIT:ANGL DEGR"  
                              OUTPUT 711;"calculate2:unit:angl rad"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:ANGLE?

Return Format:             DEGR|RAD

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   DEGR  
                              SCPI Compliance:   confirmed

Description:

This command is only valid when phase trace coordinates are specified (CALC:FORM PHAS or CALC:FORM UPH).

To select phase units in degrees for the specified trace, send CALC:UNIT:ANGL DEGR. To select phase units in radians for the specified trace, send CALC:UNIT:ANGL RAD.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:UNIT:DBReference****command/query**

Specifies the reference for dB magnitude trace coordinates.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT DBReference <UNIT>

```
<UNIT> ::= 'DBV' | "DBV"
          'DBM' | "DBM"
          'DBSPL' | "DBSPL"
          'DBUSER' | "DBUSER"
```

Example Statements:       OUTPUT 711;"CALC:UNIT:DBREFERENCE 'DBM' "  
                          OUTPUT 711;"calc3:unit:dbr 'DBUSER' "

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :UNIT:DBReference?

Return Format:            "DBV" | "DBM" | "DBSPL" | "DBUSER"

Attribute Summary:       Option: not applicable  
                          Synchronization Required: yes  
                          Preset State: "DBV"  
                          SCPI Compliance: instrument-specific

## Description:

This command allows you to scale the dB magnitude based on the parameter you select. The setting applies only for the current measurement data and the specified trace. The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

Send CALC:UNIT:DBR 'DBV' to reference the dB magnitude to 1 volt. This is the default selection.

Send CALC:UNIT:DBR 'DBM' to reference the dB magnitude to 1 milliwatt. Use the CALC:UNIT:DBR:IMPedance command to specify an impedance value that matches the impedance of your system under test. If transducer units are enabled, DBM units are valid only when the transducer unit label is volts (SENSE:VOLTAGE:RANGE:UNIT:USER:LABEL 'V').

Send CALC:UNIT:DBR 'DBSPL' to set the dB magnitude reference level to 20  $\mu$ Pa. An engineering unit of Pascals (Pa) must be applied to the data. See the SENSE:VOLTAGE:RANGE:UNIT:USER commands for additional information about enabling transducer units.

Send CALC:UNIT:DBR 'DBUSER' to set your own dB magnitude reference level. Use the CALC:UNIT:DBR:USER:REference command to specify the reference level. Use the CALC:UNIT:DBR:USER:LABel command to assign a name to the Y- axis unit.

The dB magnitude reference level is only applied to traces with dB magnitude coordinates (CALC:FORM MLOG). The dB reference scaling is applied after transducer units have been applied.

**CALCulate[1 | 2 | 3 | 4]:UNIT:DBReference:IMPedance****command/query**

Specifies the system's reference impedance value in ohms ( $\Omega$ ).

Command Syntax:            `CALCulate [1|2|3|4]:UNIT:DBReference:IMPedance {<number> [ <unit> ] } | <step> | <bound>`

`<number> ::= a real number (NRf data)  
                  limits: 1e-15:1e+15`

`<unit> ::= [OHM]`

`<step> ::= UP|DOWN`

`<bound> ::= MAX|MIN`

Example Statements:        `OUTPUT 711;" :calc2:unit:dbr:impedance 600"`  
                              `OUTPUT 711;"CALC:UNIT:DBR:IMP 3.2e3 ohm"`

Query Syntax:              `CALCulate [1|2|3|4]:UNIT:DBReference:IMPedance?`

Return Format:              `Real`

Attribute Summary:        `Option: not applicable`  
                              `Synchronization Required: yes`  
                              `Preset State: 50 OHM`  
                              `SCPI Compliance: instrument-specific`

## Description:

This command specifies the system reference impedance for the dBm reference level. The dBm unit is referenced to 1 milliwatt. Specify a value that matches the impedance of the system under test.

For example, the system impedance of a telephone system is typically 600  $\Omega$ .

The trace specifier, CALCulate [1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:UNIT:DBReference:USER:LABel****command/query**

Assigns a name to the Y-axis unit when CALC:UNIT:DBR 'USER'.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:DBReference:USER:LABel '<NAME>'

<NAME> ::= ASCII characters - 32 through 126  
          maximum number of characters: 5

Example Statements:        OUTPUT 711;"CALC:UNIT:DBR:USER:LAB '>' "  
                              OUTPUT 711;"calculate2:unit:dbreference:user:label  
                              'm/s^2' "

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:DBReference:USER:LABel?

Return Format:             "STRING"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: yes  
                              Preset State: "V"  
                              SCPI Compliance: instrument-specific

## Description:

The name assigned with this command labels the display's Y- axis. A prefix of "dB" is attached to the name. The label appears only in dB magnitude trace coordinates (CALC:FORM MLOG) and when the user dB magnitude reference level has been specified with the CALC:UNIT:DBR 'USER' command.

The trace specifier, CALCulate [1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.



**CALCulate[1 | 2 | 3 | 4]:UNIT:DBReference:USER:REference****command/query**

Specifies the reference level for CALC:UNIT:DBR 'USER'.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:DBReference:USER:REference  
                              <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                              limits: 1e-15:1e+15

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:        OUTPUT 711; ":calculate:unit:dbr:user:ref 20.0"  
                              OUTPUT 711; "CALC4:UNIT:DBR:USER:REF 1e3"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:DBReference:USER:REference?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: yes  
                              Preset State: 1.0  
                              SCPI Compliance: instrument-specific

**Description:**

Use this command to specify your own dB magnitude reference level. Send CALC:UNIT:DBR 'USER' to enable your own dB reference level. Use the CALC:UNIT:DBR:USER:LABEL command to assign a name to the Y-axis unit.

The trace specifier, CALCulate [1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:UNIT:MECHanical****command/query**

Converts the Y-axis trace coordinates from the input's transducer units to the selected engineering units on the display.

Command Syntax:            CALCulate [1|2|3|4]:UNIT:MECHanical <UNIT>

```
<UNIT> ::= `G'|"G"
          `M/S2'|"M/S2"
          `M/S'|"M/S"|
          `M'|"M"
          `INCH/S2'|"INCH/S2"
          `INCH/S'|"INCH/S"
          `INCH'|"INCH"
          `MILS'|"MILS"
```

Example Statements:        OUTPUT 711;"CALC2:UNIT:MECH `M/S' "  
                              OUTPUT 711;"calculate:unit:mechanical:volt `mils' "

Query Syntax:             CALCulate [1|2|3|4]:UNIT:MECHanical?

Return Format:            "G"|"M/S2"|"M/S"|"M"|"INCH/S2"|"INCH/S"|"INCH"|"MILS"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

## Description:

The analyzer integrates or differentiates the displayed trace coordinates to convert the display to the specified unit.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

---

**Note**                    This command is only valid when the transducer units are G's, m/S<sup>2</sup>, m/S, m, inch/S, inch/S, inch, or mils. See the SENSE:VOLTAGE:DC:RANGE:UNIT:XDCR:LABEL command for information about setting transducer units.

---

**CALCulate[1 | 2 | 3 | 4]:UNIT:VOLTage****command/query**

Selects the vertical unit for the specified display's Y-axis.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:VOLTage <UNIT>

```
<UNIT> ::=  'V' | "V"
           'V2' | "V2"
           'V/RTHZ' | "V/RTHZ"
           'V2/HZ' | "V2/HZ"
           'V2S/HZ' | "V2S/HZ"
```

Example Statements:        OUTPUT 711;"CALC2:UNIT:VOLT 'V' "  
                              OUTPUT 711;"calculate:unit:volt 'V2/HZ' "

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:VOLTage?

Return Format:             "V" | "V2" | "V/RTHZ" | "V2/HZ" | "V2S/HZ"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   "V2"  
                              SCPI Compliance:   instrument-specific

Description:

With some measurements, you can select the unit for the Y- axis scale.

- To select volts, send CALC:UNIT:VOLT 'V'.
- To select volts<sup>2</sup>, send CALC:UNIT:VOLT 'V2'. This is the default selection.
- To select square root power spectral density, send CALC:UNIT:VOLT 'V/RTHZ'.
- To select power spectral density, send CALC:UNIT:VOLT 'V2/HZ'.
- To select energy spectral density, send CALC:UNIT:VOLT 'V2S/HZ'.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

Depending upon the measurement data selection and specified trace coordinates, the selection of the base unit may be restricted. In addition, the analyzer does not permit specification of the vertical unit for some types of measurement data.

Table 6-7 indicates valid unit selections for the CALC:UNIT:VOLT command. If measurement data does not appear in the table, you are not permitted to select the base unit. In this case, a query returns a null string.

See "Determining Units" in Appendix E for information about available Y-axis units.

Table 6-7. Valid unit selections for CALC:UNIT:VOLT

Measument Data CALC:FEED command (INST:SEL command)	CALC:UNIT:VOLT				
	V	V2	V/RTHZ	V2/HZ	V2S/HZ
Composite Power CALC:FEED 'XFR:POW:COMP' (INST:SEL ORD)	X	X			
Linear Spectrum CALC:FEED 'XFR:POW:LIN' (INST:SEL FFT)	X	X	X	X	X
Linear Spectrum CALC:FEED 'XFR:POW:LIN' (INST:SEL SINE)	X	X			
Order Track CALC:FEED 'XORD:TRACK' (INST:SEL ORD)	X	X			
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL FFT)	X	X	X	X	X
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL OCT)	X	X	X	X	X
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL ORD)	X	X			

**CALCulate[1 | 2 | 3 | 4]:UNIT:X****command/query**

Specifies the X-axis unit.

Command Syntax:            CALCulate [1 | 2] :UNIT:X <UNIT>

```
<UNIT> ::= 'HZ' | "HZ"
          'CPM' | "CPM"
          'ORD' | "ORD"
          'USER' | "USER"
```

Example Statements:        OUTPUT 711;"CALC:UNIT:X 'RPM' "  
                              OUTPUT 711;"calculate2:unit:x 'USER' "

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:X?

Return Format:             "HZ | "CPM" | "ORD" | "USER"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   "HZ"  
                              SCPI Compliance:   instrument-specific

Description:

Send CALC:UNIT:X 'HZ' to specify Hertz for frequency domain X-axis units and seconds for time domain X-axis units.

Send CALC:UNIT:X 'CPM' to specify CPM for frequency domain X-axis units and seconds for time domain X-axis units.

Send CALC:UNIT:X 'ORD' to specify orders for frequency domain X-axis units and revolutions for time domain X-axis units. Use the CALC:UNIT:X:ORDER:FACTor command to specify the Hertz/Order or RPM/Order ratio.

Send CALC:UNIT:X 'USER' to specify your own X-axis units. Use the CALC:UNIT:X:USER:FREQuency:FACTor or the CALC:UNIT:X:USER:TIME:FACTor commands to specify the conversion factor. You can specify a name for the units with the CALC:UNIT:X:USER:FREQuency:LABel or the CALC:UNIT:X:USER:TIME:LABel commands.

## CALCulate

This command is valid for all measurement data selections in the FFT, swept sine and correlation instrument modes. It is also valid for time data in the histogram instrument mode. It is not available for measurement data selections in the octave analysis nor the order analysis instrument modes.

The X-axis unit is applied to the specified trace. Omit the specifier or send CALC1 for trace A, CALC2 for trace B, CALC3 for trace C, or CALC4 for trace D.

The X-axis unit setting applies to the measurement data selection independent of the instrument mode. For example, if you specify the X-axis unit for time data in the FFT instrument mode, that X-axis unit is applied to time data in the correlation instrument mode as well.

The X-axis unit is not applied to measurement data stored in any of the analyzer's registers (data, waterfall or math). To set the X-axis unit for measurement data stored in a register, display the register data in one of the trace boxes. Send the command specifying that trace.

**CALCulate[1 | 2 | 3 | 4]:UNIT:X:ORDER:FACTOR****command/query**

Specifies the speed of rotation in Hertz per Order or RPM per Order.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:X:ORDER:FACTOR {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 1e-15:1e+15

<unit> ::= [HZ/ORD|RPM/ORD]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"CALC:UNIT:X:ORD:FACT 600 RPM/ORD"  
                              OUTPUT 711;"calc3:unit:x:order:factor 10 hz/ord"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:X:ORDER:FACTOR?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: yes  
                              Preset State: 1 HZ/ORD  
                              SCPI Compliance: instrument-specific

## Description:

Use this command to specify the speed of rotation when specifying orders (or revolutions for time domain traces) as the X-axis unit (CALC:UNIT:X 'ORD').

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

To determine the unit, send CALC:UNIT:X:ORD:FACT? UNIT.

**CALCulate[1 | 2 | 3 | 4]:UNIT:X:USER:FREQUENCY:FACTOR****command/query**

Specifies the frequency conversion factor for user-defined X-axis units.

**Command Syntax:**            `CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:FREQUENCY:FACTOR <number> | <step> | <bound>`

`<number> ::= a real number (NRf data)  
          limits: 1e-15:1e+15`

`<step> ::= UP|DOWN`

`<bound> ::= MAX|MIN`

**Example Statements:**        `OUTPUT 711;"CALC:UNIT:X:USER:FREQUENCY:FACTOR 1.667e-2"`  
                              `OUTPUT 711;"calc:unit:x:user:freq:fact 0.159"`

**Query Syntax:**              `CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:FREQUENCY:FACTOR?`

**Return Format:**              `Real`

**Attribute Summary:**        `Option: not applicable`  
                              `Synchronization Required: yes`  
                              `Preset State: 1`  
                              `SCPI Compliance: instrument-specific`

**Description:**

Use this command with the CALC:UNIT:X 'USER' command for frequency domain traces.

The value you specify is entered as the number of Hertz per X-axis unit. For example, if the unit is 'cpm', the value accompanying CALC:UNIT:X:USER:FREQ:FACTOR is interpreted as Hertz/cpm.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.



**CALCulate[1 | 2 | 3 | 4]:UNIT:X:USER:FREQuency:LABel** **command/query**

Assigns a name to the user-defined X-axis units in the frequency domain.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:FREQuency:LABel '<STRING>'

Example Statements:        OUTPUT 711;"CALC2:UNIT:X:USER:FREQ:LAB '<string>'"  
                               OUTPUT 711;"calc:unit:x:user:frequency:label '<string>'"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:FREQuency:LABel?

Return Format:             "<STRING>"

Attribute Summary:        Option:   not applicable  
                               Synchronization Required:   yes  
                               Preset State:   'Hz'  
                               SCPI Compliance:   instrument-specific

**Description:**

Use this command with the CALC:UNIT:X 'USER' command for frequency domain traces.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:UNIT:X:USER:TIME:FACTOR****command/query**

Specifies the time conversion factor for user-defined X-axis units.

Command Syntax:            CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:TIME:FACTOR <number> |  
                              <step> | <bound>

<number> ::= a real number (NRf data)  
                              limits: 1e-15:1e+15

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"CALC:UNIT:X:USER:TIME:FACT 1.0789e3"  
                              OUTPUT 711;"calculate2:unit:x:user:time:factor 331.45"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :UNIT:X:USER:TIME:FACTOR?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: yes  
                              Preset State: 1  
                              SCPI Compliance: instrument-specific

**Description:**

Use this command with the CALC:UNIT:X 'USER' command for time domain traces.

The value you specify is entered as the number of X-axis units per second. For example, if the unit is 'ft', the value accompanying CALC:UNIT:X:USER:TIME:FACTOR is interpreted as ft/second.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:UNIT:X:USER:TIME:LABel****command/query**

Assigns a name to the user-defined X-axis units in the time domain.

Command Syntax:           CALCulate [1|2|3|4]:UNIT:X:USER:TIME:LABel '<STRING>'

Example Statements:        OUTPUT 711;"CALCULATE2:UNIT:X:USER:TIME:LABEL 'ft' "  
                              OUTPUT 711;"CALC:UNIT:X:USER:TIME:LAB 'm' "

Query Syntax:             CALCulate [1|2|3|4]:UNIT:X:USER:TIME:LABel?

Return Format:             "<STRING>"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   's'  
                              SCPI Compliance:   instrument-specific

**Description:**

Use this command with the CALC:UNIT:X 'USER' command for time domain traces.

The trace specifier, CALCulate[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

**CALCulate[1 | 2 | 3 | 4]:WATERfall:COUNT****command/query**

Specifies the number of traces stored for waterfall displays.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :WATERfall:COUNT <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 1:32767

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:       OUTPUT 711;":calculate:waterfall:count 50"  
                          OUTPUT 711;"CALC2:WAT:COUN 32"

Query Syntax:            CALCulate [1 | 2 | 3 | 4] :WATERfall:COUNT?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +15  
                          SCPI Compliance: instrument-specific

**Description:**

This command determines the total capacity of waterfall displays. The capacity of a waterfall display is determined by the number of complete measurements that are stored in memory. This in turn, determines the number of traces that appear in a waterfall display.

When you change instrument modes or start a new measurement with the ABOR;:INIT command, all current waterfall traces are lost.

---

**Note**                    The maximum number of traces stored for a waterfall display is dependent upon the amount of available memory. This command ignores the trace specifier.

---

For more information, see online help for the [WATERFALL STEPS] softkey.

**CALCulate[1 | 2 | 3 | 4]:WATERfall[:DATA]?****query**

Returns waterfall data that has been transformed to the currently selected coordinate transform (specified with CALC:FORMat).

Query Syntax:                    CALCulate [1 | 2 | 3 | 4] :WATERfall [:DATA] ?

Example Statements:            OUTPUT 711; ":Calculate:Wat?"  
                                   OUTPUT 711; "CALCULATE:WAT:DATA?"

Return Format:                   <BLOCK>

If FORMat[:DATA] REAL:

```
<BLOCK> ::= #<byte>[<length_bytes>] <1st_waterfall_value>
           . . . <last_waterfall_value>
<byte>   ::= one byte specifying the number of length bytes
           to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)
```

If FORMat[:DATA] ASCii:

```
<BLOCK> ::= <1st_waterfall_value> . . . <last_waterfall_value>
```

Attribute Summary:            Option:   not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns a definite length block of coordinate-transformed waterfall data.

The trace specifier CALCulate [1|2|3|4] determines which waterfall you are selecting. Omit the specifier or send 1 for trace box A, 2 for trace box B, 3 for trace box C, or 4 for trace box D.

A waterfall consists of a series of traces. To determine the number of traces in the waterfall, divide the number of values in the waterfall (<length\_bytes> / 8) by the number of values in the trace (CALC:DATA:HEADer:POINts?).

The block is returned as a series of amplitude values (<1st\_trace\_1st\_Y-axis\_value>, . . . <last\_trace\_last\_Y-axis\_value>). The unit for these values is the same as the reference level unit. To determine the unit, send DISP:TRAC:Y:BOTT? UNIT.

For orbit diagrams (CALC:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]') the block is a series of real-value pairs (<1st\_trace\_1st\_X-axis\_value>, <1st\_trace\_1st\_Y-axis\_value>, . . . <last\_trace\_last\_X-axis\_value>, <last\_trace\_last\_Y-axis\_value>).

For Nyquist diagrams (CALC:FORM NYQ) the block is a series of real and imaginary pairs (<1st\_trace\_1st\_real\_value>, <1st\_trace\_1st\_imaginary\_value>, . . . <last\_trace\_last\_real\_value>, <last\_trace\_last\_imaginary\_value>).

## CALCulate

For polar diagrams (CALC:FORM POL) the block is a series of magnitude and phase pairs (<1st\_trace\_1st\_magnitude\_value>, <1st\_trace\_1st\_phase\_value>, . . . <last\_trace\_last\_magnitude\_value>, <last\_trace\_last\_phase\_value>).

In octave analysis instrument mode (INST:SEL OCT;Option 1D1) the second to the last value in the trace data is the weighted overall band. The last value in the trace data is the overall band. These values (for each trace) are included in the data block, even if they are not displayed.

Use the CALC:X:DATA? query to determine the X-axis values for the waterfall data. Use the TRACe:Z[:DATA]? query to determine the Z-axis values for the waterfall data.

This query has no command form. You cannot return waterfall data to the display with CALC:WAT:DATA. To send data that has not been transformed, use the TRAC:WAT[:DATA] command. See the introduction to this chapter for more information about the differences between these commands.

---

**CALCulate[1 | 2 | 3 | 4]:WATERfall:SLICe:COPI****command**

Copies the selected waterfall slice to the designated data register.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :WATERfall:SLICe:COPI  
                              D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Example Statements:        OUTPUT 711;"calculate4:wat:slice:copy D1"  
                              OUTPUT 711;"Calc3:Waterfall:Slic:Copy D8"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

A slice is a vertical line through the collection of waterfall traces at the same X- axis value. This command copies the slice to the specified data register. Use the CALC:WAT:SLICe:SELEct command to select the slice.

**CALCulate[1 | 2 | 3 | 4]:WATERfall:SLICE:SElect****command/query**

Selects the waterfall slice at the specified X-axis position.

Command Syntax:       CALCulate [1 | 2 | 3 | 4] :WATERfall:SLICE:SElect {<number>  
                          [<unit>]} |<step> |<bound>

<number> ::= a real number (NRf data)  
                  limits: -16382:1e6

<unit> ::= [HZ | S | ORD | COUNT | AVG | V | VPK | RPM | EU | REV]

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements:     OUTPUT 711;" :CALC4:WAT:SLICE:SEL 880661"  
                          OUTPUT 711;"calculate:wat:slic:select 757455"

Query Syntax:           CALCulate [1 | 2 | 3 | 4] :WATERfall:SLICE:SElect?

Return Format:          Real

Attribute Summary:     Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

## Description:

This command specifies the X-axis position where the waterfall slice is to be made. The slice can be copied to a data register with the CALC:WAT:SLIC:COPY command.

To specify a waterfall trace in trace box A, send CALC1; in trace box B, send CALC2; in trace box C, send CALC3; and in trace box D, send CALC4. The trace specifier defaults to trace box A if the specifier is not used.



**CALCulate[1 | 2 | 3 | 4]:WATERfall:SLICe:SElect:POINt****command/query**

Selects a waterfall slice by its display point value.

Command Syntax: `CALCulate [1 | 2 | 3 | 4] :WATERfall:SLICe:SElect:POINt <number> | <step> | <bound>`

`<number> ::= a real number (NRf data)  
limits: 0:2048`

`<step> ::= UP | DOWN`

`<bound> ::= MAX | MIN`

Example Statements: `OUTPUT 711; "Calc:Waterfall:Slic:Sel:Point 816"`  
`OUTPUT 711; "CALC4:WATERFALL:SLIC:SEL:POINT 1409"`

Query Syntax: `CALCulate [1 | 2 | 3 | 4] :WATERfall:SLICe:SElect:POINt?`

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

**Description:**

This command specifies the X-axis position for the waterfall slice by point number.

The number of points displayed along the X-axis depends upon the number of lines of resolution set with the [SENSe:]FREQuency:RESolution command. See table 6-8.

In correlation analysis the number of points displayed along the X-axis depends upon the resolution and the windowing function. See table 6-8.

In histogram analysis, the number of points is determined by the number of bins (set with the [SENS:]HIST:BINS command). The maximum number is 1024 points.

In octave analysis, the number of points is determined by the bandwidth of the filters. There are 11 points for full octave, 33 points for 1/3 octave and 132 points for 1/12 octave.

In some cases the number of points is arbitrary. These include waterfall displays from order tracking or from the arbitrary source.

See the CALC:WAT:SLICe:COpy command for information about saving a waterfall slice to a data register.

Table 6-8. Number of displayed data points in variable resolution

## FFT Instrument Mode

Resolution	Baseband		Zoom (Start frequency $\sqrt{0}$ )	
	Number of frequency points (frequency data)	Number of real time points (time data)	Number of frequency points (frequency data)	Number of complex time points (real/ imaginary pairs) (time data)
100	101	256	101	128
200	201	512	201	256
400	401	1024	401	512
800	801	2048	801	1024

Swept Sine Instrument Mode  
(FREQ:RES:AUTO ON)

FREQ:RES:AUTO:MIN (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to FREQ:SPAN / FREQ:RES:AUTO:MIN. With PCT, the spacing between measurement points is a percentage of the total frequency span.

---

**CALCulate[1 | 2 | 3 | 4]:WATERfall:TRACe:COPI****command**

Saves the selected trace to the specified data register.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :WATERfall:TRACe:COPI  
                              D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8

Example Statements:        OUTPUT 711;":calc3:waterfall:trac:copy D7"  
                              OUTPUT 711;"Calculate:Wat:Trace:Copy D4"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

This command copies a trace, selected with the CALC:WAT:TRACe:SElect command, from the waterfall to the specified data register.

See the CALC:WAT:SLICe commands for information about selecting and saving waterfall slices.

**CALCulate[1 | 2 | 3 | 4]:WATERfall:TRACe:SElect****command/query**

Selects a waterfall trace by its Z-axis value.

Command Syntax:            CALCulate [1 | 2 | 3 | 4] :WATERfall:TRACe:SElect {<number>  
                                  [<unit>]} |<step> |<bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37

<unit> ::= [S|RPM|COUNT|AVG]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"calculate:wat:trace:sel 10 count"  
                                  OUTPUT 711;"CALC3:WAT:TRAC:SEL 3.5 S"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :WATERfall:TRACe:SElect?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                                  Synchronization Required: no  
                                  Preset State: +0.0  
                                  SCPI Compliance: instrument-specific

## Description:

This command selects a waterfall trace by its Z-axis value. The Z-axis value indicates when the measurement data was armed.

To specify a waterfall trace in trace box A, send CALC1; in trace box B, send CALC2; in trace box C, send CALC3; and in trace box D, send CALC4. The trace specifier defaults to trace box A if the specifier is not used.

See the CALC:WAT:TRAC:COPY command for information about copying a selected trace to a data register.

**CALCulate[1 | 2 | 3 | 4]:WATERfall:TRACe:SElect:POINT****command/query**

Selects a waterfall trace by its step value.

Command Syntax:           CALCulate [1 | 2 | 3 | 4] :WATERfall:TRACe:SElect:POINT <number>  
                                  | <step> | <bound>

<number> ::= a real number (NRf data)  
                                  limits: 0:32767

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"CALC4:WATERFALL:TRAC:SELECT:POIN 4104"  
                                  OUTPUT 711;"calc4:waterfall:trac:select:poin 15879"

Query Syntax:             CALCulate [1 | 2 | 3 | 4] :WATERfall:TRACe:SElect:POINT?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

**Description:**

This command selects a waterfall trace by its step value. The total number of waterfall steps is specified with the CALC:WAT:COUNt command.

A value of 1 specifies the first trace collected for the waterfall display. A value of 2 specifies the second trace collected for the waterfall display.

To specify a waterfall trace in trace box A, send CALC1; in trace box B, send CALC2; in trace box C, send CALC3; and in trace box D, send CALC4. The trace specifier defaults to trace box A if the specifier is not used.

See the CALC:WAT:TRAC:COPY command for information about copying a selected trace to a data register.

**CALCulate[1 | 2 | 3 | 4]:X:DATA?****query**

Returns the X-axis values that correspond to the Y-axis values read with the CALC:DATA? query.

Query Syntax:                   CALCulate [1 | 2 | 3 | 4] :X:DATA?

Example Statements:            OUTPUT 711;" :calc:x:data?"  
                                   OUTPUT 711;"Calculate:X:Data?"

Return Format:                   <DEF\_BLOCK>

```
If FORMat [:DATA] REAL:

<DEF_BLOCK> ::= #<byte><length_bytes> <1st_X-axis_value> . . .
               <last_X-axis_value>
    <byte> ::= number of length bytes to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)

If FORMat [:DATA] ASCII:

<DEF_BLOCK> ::= <1st_X-axis_value> . . .<last_X-axis_value>
```

Attribute Summary:            Option:   not applicable  
                                   Synchronization Required: no  
                                   Preset State:   not applicable  
                                   SCPI Compliance: confirmed

## Description:

This query returns a definite length block of the X-axis values that correspond to coordinate-transformed trace data obtained using the CALC:DATA? query.

The trace specifier CALCulate [1|2|3|4] determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

The block is returned as a series of X-axis values. To determine the unit, send CALC:MARKer:POSition? UNIT.

For orbit diagrams (CALC:FEED 'XVOL:VOLT [1,2|1,3|1,4|3,4]') the block consists of the time values (T) for the real-value pairs (<1st\_T\_value> for <1st\_X-axis\_value>, <1st\_Y-axis\_1\_value>, . . . <last\_T\_value> for <last\_X-axis\_value>, <last\_Y-axis\_value>).

For Nyquist diagrams (CALC:FORM NYQ) the block consists of the frequency values for the real and imaginary pairs.

For polar diagrams (CALC:FORM POL) the block consists of the frequency values for the magnitude and phase pairs.

In octave analysis instrument mode (INST:SEL OCT;Option 1D1) or for octave data stored in data registers, the last two values are 9.1E37. They correspond to the X-axis values for the weighted overall band and the overall band. These values are meaningless.

This query has no command form.

## CAlibration

---





## CAlibration

---

This subsystem contains commands related to calibration of the analyzer.

**CALibration[:ALL]?**

**query**

Calibrates the analyzer and returns the result.

Query Syntax:           CALibration[:ALL]?

Example Statements:     OUTPUT 711;":Cal?"  
                          OUTPUT 711;"CALIBRATION:ALL?"

Return Format:           Integer

Attribute Summary:     Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

The analyzer performs a full calibration when you send this query. If the calibration completes without error, the analyzer returns 0. If the calibration fails, the analyzer returns 1.

This query is the same as the \*CAL? query.

---

**CALibration:AUTO****command/query**

Calibrates the analyzer or sets the state of the autocalibration function.

Command Syntax:           CALibration:AUTO OFF|0|ON|1|ONCE

Example Statements:        OUTPUT 711;"calibration:auto OFF"  
                              OUTPUT 711;"Cal:Auto ONCE"

Query Syntax:             CALibration:AUTO?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  ON (+1)  
                              SCPI Compliance:  confirmed

Description:

Send CAL:AUTO ON to enable the analyzer's autocalibration function, OFF to disable it. This function calibrates the analyzer several times during the first hour of operation and once per 140 minutes thereafter.

Send CAL:AUTO ONCE to initiate a single calibration.

---

**Note**                    CAL:AUTO is set to +0 (OFF) after \*RST.

---



DISPly

---



## DISPlay

---

This subsystem contains commands that control the analyzer's presentation of data on its front-panel display.

The DISPlay subsystem contains commands grouped under the WINDow keyword. The WINDow keyword contains an optional trace specifier: [1|2|3|4]. To direct a command to trace A, omit the specifier or use WIND1. To direct a command to trace B, use WIND2; to trace C, use WIND3; and to trace D, use WIND4.

WINDow is an implied mnemonic. Therefore, you can omit it from DISPlay commands. However, if you wish to direct a DISPlay command to a specific trace box, you must use the WINDow trace specifier. See "Implied Mnemonics" in chapter 2 of the GPIB Programmer's Guide for more information.

**DISPlay:ANNotation[:ALL]****command/query**

Turns the display of screen annotation on or off. \$Idisplay;disabling annotation

Command Syntax: DISPlay:ANNotation[:ALL] OFF|0|ON|1

Example Statements: OUTPUT 711;":DISP:ANNOTATION ON"  
OUTPUT 711;"disp:ann:all OFF"

Query Syntax: DISPlay:ANNotation[:ALL]?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: ON (+1)  
SCPI Compliance: confirmed

**Description:**

When DISP:ANN is OFF, the following information is *not* displayed on the analyzer's screen:

- X-axis annotation
- Y-axis annotation
- Z-axis annotation
- Marker annotation
- Mini-state

In addition, this information does not appear in a plot or print of the screen if DISP:ANN is OFF. It is available, however, to GPIB queries.

When DISP:ANN is ON, all information, including the annotation, is displayed on the analyzer's screen. This is the default setting.



---

**DISPlay:BODE****command**

Displays a Bode diagram.

Command Syntax:           DISPlay:BODE

Example Statements:        OUTPUT 711;"Disp:Bode"  
                              OUTPUT 711;"DISP:BODE"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The Bode diagram formats the display as follows:

- The frequency response of the measurement data.
- The trace coordinate for trace box A and trace box C is phase.
- The trace coordinate for trace box B and trace box D is dB magnitude.
- The X-axis scale is logarithmic.
- The markers are coupled.

---

**Note**                    To change the X-axis to linear scaling, send the  
                              DISPlay:WINDow[1|2|3|4]:TRACe:X:SPACing LINear command.

---

**DISPlay:BRIGhtness****command/query**

Adjusts the intensity of the display.

Command Syntax:           DISPlay:BRIGhtness <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0.5:1  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;"disp:brig 0.759"  
                          OUTPUT 711;"DISPLAY:BRIGHTNESS 1"

Query Syntax:             DISPlay:BRIGhtness?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: 1  
                          SCPI Compliance: confirmed

Description:

To specify full intensity, send DISP:BRIG 1.

To specify medium intensity, send DISP:BRIG 0.5.

To turn off or to disable the display, use the DISP:STATE OFF command.

**DISPlay:ERRor****command**

Displays text in the same format as the analyzer.

Command Syntax:           DISPlay:ERRor '<STRING>'

<STRING> ::= ASCII characters - 0 through 255  
          maximum number of characters: 32766

Example Statements:       OUTPUT 711;"DISP:ERR 'Please try again.'"  
                          OUTPUT 711;"display:error 'Enter another value.'"

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

The analyzer displays error messages in a pop-up message window at the center of the screen. The message window appears on the screen for approximately 7 seconds.

This command allows you to display an error message in the same manner as the analyzer.

**DISPlay:EXTernal[:STATe]****command/query**

Enables the use of an external monitor.

Command Syntax:           DISPlay:EXTernal[:STATe] OFF|0|ON|1

Example Statements:       OUTPUT 711;":display:ext ON"  
                          OUTPUT 711;"Display:Ext:Stat OFF"

Query Syntax:             DISPlay:EXTernal[:STATe]?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0 (OFF)  
                          SCPI Compliance: instrument-specific

**Description:**

Send DISP:EXT ON, to display the front panel screen on an external monitor. The front panel display is active while DISP:EXT is ON. However, the brightness of the front panel display is reduced by 30 percent from its maximum.

For more information about setting up an external monitor, see online help.

**DISPlay:FORMat****command/query**

Selects a format for displaying trace data.

Command Syntax:           DISPlay:FORMat SINGle|ULOWer|FBACk|SUBL|ULFB|QUAD

Example Statements:       OUTPUT 711;"DISPLAY:FORM ULFB"  
                               OUTPUT 711;"display:form ULOWER"

Query Syntax:             DISPlay:FORMat?

Return Format:             SING|ULOW|FBAC|SUBL|ULFB|QUAD

Attribute Summary:       Option: not applicable  
                               Synchronization Required: no  
                               Preset State: SING  
                               SCPI Compliance: instrument-specific

## Description:

When you select SING, the analyzer uses the entire screen for the active trace box. If multiple traces are active (set with the CALC:ACTIVE command), the active traces are changed to a single active trace—trace A or trace C.

When you send ULOW, the analyzer uses the upper half of the screen for trace box A or trace box C and the lower half for trace box B or trace box D.

When you select FBAC, the analyzer uses the entire screen, but overlays the two trace boxes in the same area.

When you select SUBL, the analyzer uses the upper quarter of the screen for trace box A or trace box C and the lower portion of the screen for trace box B or trace box D.

When you send ULFB, the analyzer combines the ULOW format with the FBAC format. Trace A is the front and trace B is the back in the upper trace box. Trace C is the front and trace D is the back in the lower trace box.

When you send QUAD, the analyzer displays four trace boxes.

---

**Note**                   Waterfall displays (DISP[:WIND]:WAT[:STATE] ON) are not valid in FBACk or ULFB formats.

---

**DISPlay:GPIB:ECHO****command/query**

Enables and disables the echoing of GPIB command mnemonics to the analyzer's screen.

Command Syntax:           DISPlay:GPIB:ECHO OFF|0|ON|1

Example Statements:        OUTPUT 711;":Disp:Gpib:Echo OFF"  
                              OUTPUT 711;"DISPLAY:GPIB:ECHO ON"

Query Syntax:             DISPlay:GPIB:ECHO?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: OFF (+0)  
                              SCPI Compliance: instrument-specific

**Description:**

When echoing is enabled, the analyzer displays the GPIB command mnemonic which corresponds to the operation executed from the front panel. The command mnemonic appears on the third line in the upper-left corner of the screen.

Not every keystroke generates an GPIB command.

**DISPlay:PROG:KEY:BOX****command/query**

Draws a box around softkey in an Instrument BASIC program.

Command Syntax:           DISPlay:PROG:KEY:BOX <key\_number>,OFF|0|ON|1

```
<key_number> ::= <number>|<bound>
<number> ::= a real number (NRf data)
             limits: 0:8
<bound> ::= MAX|MIN
```

Example Statements:       OUTPUT 711;":disp:prog:key:box 1,ON"  
                          OUTPUT 711;":DISPLAY:PROGRAM:KEY:BOX 3,OFF"

Query Syntax:

```
DISPlay:PROG:KEY:BOX? <key_number>
```

Return Format:           Integer, Integer

Attribute Summary:       Option: IC2 Instrument BASIC  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

Use this command to make the softkeys created by an Instrument BASIC program, appear similar to the analyzer's softkeys.

The first parameter specifies which softkey is enclosed in a box. The valid range is 0 to 8; with 0 representing the top softkey and 8 representing the bottom softkey.

The second parameter turns on or turns off the softkey box.

See the ON KEY command in the Instrument BASIC User's Handbook for information on labeling softkeys.

**DISPlay:PROG:KEY:BRACKET****command/query**

Draws a bracket around one or more softkeys in an Instrument BASIC program.

Command Syntax: `DISPlay:PROG:KEY:BRACKET <first_key_number>, <last_key_number>, OFF|0|ON|1`

```
<first_key_number> ::= <number>|<bound>
<number> ::= a real number (NRf data)
              limits: 0:8
<bound> ::= MAX|MIN
<last_key_number> ::= <number>|<bound>
<number> ::= a real number (NRf data)
              limits: 0:8
<bound> ::= MAX|MIN
```

Example Statements: `OUTPUT 711;":disp:prog:key:brac 0,8,ON"`  
`OUTPUT 711;"DISPLAY:PROGRAM:KEY:BRACKET 0,4,OFF"`

Query Syntax: `DISPlay:PROG:KEY:BRACKET? <First_key_number>, <last_key_number>`

Return Format: Integer, Integer, Integer

Attribute Summary: Option: IC2 Instrument BASIC  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

**Description:**

Use this command to make the softkey menus created by an Instrument BASIC program, appear similar to the analyzer's menus. The command draws a bracket around a group of softkeys, indicating a one-of-n selection.

The first parameter specifies the first softkey to appear within the bracket. The second parameter specifies the last softkey to appear within the bracket. The valid range for both parameters is 0 to 8; with 0 representing the top softkey and 8 representing the bottom softkey.

The third parameter turns on or turns off the bracket.

You must specify the range in a query. The query only addresses the exact range—not keys within the range. For example, if the the bracket is around keys 0 through 4 (0,4), and you you sent `DISP:PROG:KEY:BRAC? 2,3`; the query would return +0 (OFF).

See the ON KEY command in the Instrument BASIC User's Handbook for information on labeling softkeys.



**DISPlay:PROGram[:MODE]****command/query**

Selects the portion of the analyzer's screen to be used for Instrument BASIC program output.

Command Syntax:           DISPlay:PROGram[:MODE] OFF|0|FULL|UPPer|LOWer

Example Statements:       OUTPUT 711;"display:prog:mode LOWER"  
                           OUTPUT 711;"Disp:Prog LOWER"

Query Syntax:             DISPlay:PROGram[:MODE] ?

Return Format:            OFF|FULL|UPP|LOW

Attribute Summary:       Option: 1C2 Instrument BASIC  
                           Synchronization Required: no  
                           Preset State: OFF (+0)  
                           SCPI Compliance: instrument-specific

Description:

FULL allocates the entire trace box for program output. UPP allocates the upper trace box. LOW allocates the lower trace box.

If DISP:PROG is OFF, the analyzer does not allocate any portion of the trace box for program output.

This command will change the display format (DISP:FORM) if the format is not compatible with the DISP:PROG selection.

**DISPlay:PROGram:VECTor:BUFFer[:STATe]****command/query**

Enables or disables the buffering of lines drawn with Instrument BASIC's graphics statements.

Command Syntax:           DISPlay:PROGram:VECTor:BUFFer[:STATe] OFF|0|ON|1

Example Statements:       OUTPUT 711;" :DISPLAY:PROG:VECTOR:BUFF OFF"  
                               OUTPUT 711;"disp:program:vect:buffer:stat OFF"

Query Syntax:             DISPlay:PROGram:VECTor:BUFFer[:STATe]?

Return Format:             Integer

Attribute Summary:       Option: 1C2 Instrument BASIC  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: instrument-specific

## Description:

This command allows you to determine if the vectors used for Instrument BASIC graphics are stored in memory.

Send DISPlay:PROGram:VECTor:BUFFer ON if you want to store all vectors created with DRAW statements in memory.

Send DISPlay:PROGram:VECTor:BUFFer OFF if you want to disable the storing of the vectors. The vectors created by DRAW statements are not stored in memory.

When lines are drawn by Instrument BASIC graphics statements, line, endpoint coordinates and pen choice information is saved in memory for every line segment drawn. The information is stored in memory so that the display can be correctly redrawn when the display partition is turned off then back on with the DISP:PROG[:MODE] command.

Using the Instrument BASIC graphics statements to create animated graphics can take up a large amount of memory. An animated graphic "moves," lines are constantly drawn, erased, and then redrawn to indicate movement. Due to the amount of memory required, you probably do not want to store vectors in an animated graphic.

**DISPlay:RPM[:STATe]****command/query**

Turns on the display of the RPM indicator.

Command Syntax:           DISPlay:RPM[:STATe] OFF|0|ON|1

Example Statements:       OUTPUT 711;"Disp:Rpm:Stat ON"  
                          OUTPUT 711;"DISPLAY:RPM OFF"

Query Syntax:             DISPlay:RPM[:STATe]?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: OFF (+0)  
                          SCPI Compliance: instrument-specific

Description:

The RPM indicator that appears at the top of the screen, displays the current RPM value of the tachometer input.

To turn off the RPM indicator, send DISP:RPM OFF.

The RPM indicator can not be turned off when in order analysis instrument mode (INST:SEL ORD) or when RPM step arming (ARM:SOUR RPM) is used.

---

**DISPlay:SHOWall[:STATe]****command/query**

Turns the show all lines feature on and off.

Command Syntax: DISPlay:SHOWall[:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;"DISP:SHOW ON"  
                              OUTPUT 711;"DISP:SHOWALL OFF"

Query Syntax:               DISPlay:SHOWall[:STATe] ?

Return Format:               Integer

Attribute Summary:         Option: not applicable  
                              Synchronization Required: no  
                              Preset state: Off  
                              SCPI Compliance: instrument-specific

**Description:**

In the FFT analysis mode, the analyzer usually shows only the alias-protected data. For example, with the frequency resolution set to 400 lines, the FFT size is 513 lines, but only the first 401 points are displayed because the last 112 lines are in the anti-alias filter transition band. When the show all lines mode is on and the anti-alias filters for all active channels are off, all the available frequency lines are displayed.

The number of lines displayed when the show all lines mode is on can be calculated by :

$$(\text{Lines of resolution}) \times 1.28 + 1$$

where lines of resolution is 100, 200, 400, 800, or 1600. See the **FREQ:RES** command for more information on lines of resolution.

Use the **INP:FILT:LPASS** command to turn the anti-alias filters off.

---

**Note**                    Measurement results are not calibrated for flatness when the anti-alias filters are off.

---

---

**DISPlay:STATe****command/query**

Turns on (or turns off) the analyzer's display.

Command Syntax:           DISPlay:STATe OFF|0|ON|1

Example Statements:        OUTPUT 711;":disp:state OFF"  
                              OUTPUT 711;"Disp:State OFF"

Query Syntax:             DISPlay:STATe?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  ON (+1)  
                              SCPI Compliance:  confirmed

Description:

To turn off the analyzer's display, send DISP:STAT OFF. All power to the display is disabled. The analyzer's screen is blanked out.

To turn on the display or to turn "display blanking" off, send DISP:STAT ON.

---

**Note**                    Press any front-panel key to turn on a blanked screen.

---

**DISPlay:TCAPture:ENVELOpe[:STATe]****command/query**

Displays the envelope of the time capture buffer.

Command Syntax:           DISPlay:TCAPture:ENVELOpe[:STATe] OFF|0|ON|1

Example Statements:       OUTPUT 711;"DISP:TCAP:ENVELOPE:STAT OFF"  
                          OUTPUT 711;"display:tcap:env OFF"

Query Syntax:             DISPlay:TCAPture:ENVELOpe[:STATe]?

Return Format:             Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: ON (+1)  
                          SCPI Compliance: instrument-specific

Description:

The envelope is a signal composed of the peak values (both positive and negative) of the original time capture data.

**DISPlay:VIEW****command/query**

Specifies what is displayed on the analyzer's screen.

Command Syntax:           DISPlay:VIEW TRACe | MSTate | MMEMemory | STABle | CTABle | FTABle |  
TTABle | MEMory | OPTion | CAPTure | ISTate | MESSage

Example Statements:       OUTPUT 711;":DISP:VIEW CAPT"  
OUTPUT 711;"display:view mstate"

Query Syntax:             DISPlay:VIEW?

Return Format:            TRAC | MST | MMEM | STAB | CTAB | FTAB | TTAB | MEM | OPT | CAPT | MESS

Attribute Summary:       Option:   not applicable  
Synchronization Required: no  
Preset State:   TRAC  
SCPI Compliance: instrument-specific

Description:

This command specifies the contents of the analyzer's display area.

- To display measurement data, send TRAC. Use the CALC:FEED commands to specify the measurement data to be displayed in the active trace.
- To display the analyzer's current measurement configuration, send MST.
- To display the contents of the default disk, send MMEM. (You select the default disk with the MMEM:MSIS command.)
- To display the synthesis table, send STAB.
- To display the curve fit table, send CTAB.
- To display the fault log table, send FTAB.
- To display the test log table, send TTAB.
- To display the memory usage table, send MEM.
- To display the option configuration of the analyzer, send OPT.
- To display header information for the time capture buffer, send CAPT.
- To display the most recent messages (up to 5) that had been displayed on the analyzer's screen, send MESS.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:DTABLE:MARKer[:STATe]****command/query**

Turns on or off the markers for data tables.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:DTABLE:MARKer[:STATe]  
                              OFF|0|ON|1

Example Statements:        OUTPUT 711;"display:wind4:dtab:marker:stat on"  
                              OUTPUT 711;"disp:dtab:mark 1"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:DTABLE:MARKer[:STATe]?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  OFF (+0)  
                              SCPI Compliance:  instrument-specific

**Description:**

Data tables have their own group of markers. They appear on the trace when the data table is turned off (DISP:DTABLE OFF).

WINDow[1|2|3|4] specifies the data table. Send WIND1 to specify the data table appearing in trace box A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional specifier, the command defaults to trace box A.



**DISPlay[:WINDow[1 | 2 | 3 | 4]]:DTABle[:STATe]****command/query**Enables and displays a data table.*ay:data tables*

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:DTABle[:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;":Display:Dtab OFF"  
                              OUTPUT 711;":DISPLAY:WIND:DTAB:STATE OFF"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:DTABle[:STATe]?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  OFF (+0)  
                              SCPI Compliance:  instrument-specific

Description:

This command turns on or off the analyzer's data tables.

WINDow[1|2|3|4] specifies the data table. Send WIND1 to specify the data table appearing in trace box A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional specifier, the command defaults to the data table appearing in trace box A.

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:LIMit:STATe****command/query**

Turns limit lines on or off in the specified display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:LIMit:STATe OFF|0|ON|1

Example Statements:        OUTPUT 711;"disp:window:lim:stat OFF"  
                          OUTPUT 711;"Display:Lim:State ON"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:LIMit:STATe?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   OFF (+0)  
                          SCPI Compliance:   instrument-specific

**Description:**

Sending DISP:LIM:STAT ON only enables the display of the limit lines in the specified trace. For example, send DISP:WIND3:LIM:STAT ON to turn on the limit line for trace C. To test the trace against those lines, you must send CALC3:LIM ON. If a trace specifier is not used, the command defaults to trace A.

---

**Note**                    A trace can be evaluated against limits even when limit lines are not displayed.

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:POLar:CLOCKwise****command/query**

Specifies the direction of the vector in a polar diagram.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:POLar:CLOCKwise OFF|0|ON|1

Example Statements:        OUTPUT 711;":DISP:POL:CLOCKWISE OFF"  
                               OUTPUT 711;"disp>window2:pol:cloc ON"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:POLar:CLOCKwise?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:  OFF (+0)  
                               SCPI Compliance:  instrument-specific

Description:

If DISP:POLar:CLOCKwise is OFF, the vector travels in a counter-clockwise direction.

If DISP:POLar:CLOCKwise is ON, the vector travels in a clockwise direction.

The setting of this command affects the DISP:POLar:ROTational command.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:POLar:ROTation****command/query**

Specifies the angle of rotation for the polar diagram.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:POLar:ROTation <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -360:+360

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Display:Wind3:Polar:Rot -311"  
                          OUTPUT 711;"DISP:POLAR:ROT -76"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:POLar:ROTation?

Return Format:           Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: 0 ° - right side of a horizontal X-axis  
                          SCPI Compliance: instrument-specific

**Description:**

This command rotates the X-axis of a polar diagram. The setting of the DISP:POLar:CLOCKwise command affects the parameter of this command. See figure 8-1.

If DISP:POLar:CLOCKwise is OFF, a positive value rotates the polar diagram up and to the left of the horizontal axis and a negative value rotates the polar diagram down and to the left.

If DISP:POLar:CLOCKwise is ON, a positive value rotates the polar diagram down and to the left of the horizontal axis and a negative value rotates the polar diagram up and to the left.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

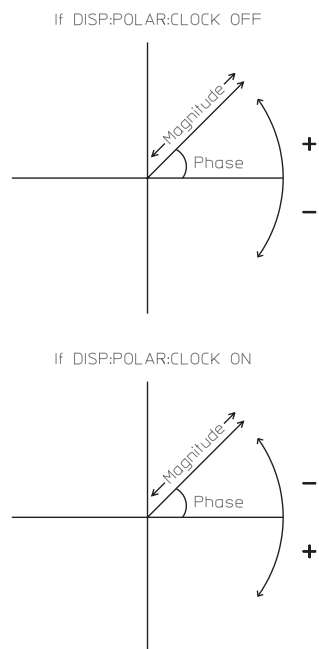


Figure 8-1. Rotation and Direction of a Polar Diagram

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:APOWer[:STATe]****command/query**

Turns on or off the display of the A-weight overall band.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:APOWer[:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;":display:trac:apow OFF"  
                               OUTPUT 711;"Display:Wind4:Trace:Apow:Stat OFF"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:TRACe:APOWer[:STATe]?

Return Format:             Integer

Attribute Summary:        Option: 1D1 Realtime Octave  
                               Synchronization Required: no  
                               Preset State: OFF (+0)  
                               SCPI Compliance: instrument-specific

## Description:

If the input channel A-weight filter is on (INPut:FILTer:AWEighting ON), it is applied for the overall band. This command enables the display of the A-weighted overall band.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

To display the overall band without the A-weight filter applied to the overall band, see the DISPlay:WINDow:TRACe:BPOWer:STATe command.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:BPOWer[:STATe]** **command/query**

Turns on or off the display of the overall band.

Command Syntax: `DISPlay[:WINDow[1|2|3|4]]:TRACe:BPOWer[:STATe] OFF|0|ON|1`

Example Statements: `OUTPUT 711;"DISPLAY:WIND:TRACE:BPOW:STAT OFF"`  
`OUTPUT 711;"display:trac:bpower ON"`

Query Syntax: `DISPlay[:WINDow[1|2|3|4]]:TRACe:BPOWer[:STATe]?`

Return Format: Integer

Attribute Summary: Option: 1D1 Realtime Octave  
 Synchronization Required: no  
 Preset State: ON (+1)  
 SCPI Compliance: instrument-specific

## Description:

This command displays the overall band in octave analysis instrument mode. A letter below the band indicates the type of power.

- T indicates band-limited total power.
- I indicates broadband impulse power (SENSe:AVERage:TYPE MAX).
- B indicates broadband total power.
- P indicates broadband peak power (SENSe:AVERage:IMPulse ON).
- U indicates undefined data (data read from a different analyzer).

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

In addition, you can display the overall band with the A-weight filter applied to the overall band. See the DISPlay:WINDow:TRACe:APOWer:STATe command for more information.

## DISPlay

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATICule:GRID[:STATe]**

**command/query**

Turns on or off the display's overlay grid.

Command Syntax:           DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATICule:GRID[:STATe]  
                          OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;":Disp:Trac:Graticule:Grid OFF"  
                          OUTPUT 711;"DISPLAY:WIND3:TRAC:GRATICULE:GRID:STATE OFF"

Query Syntax:             DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATICule:GRID[:STATe]?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:  no  
                          Preset State:  ON (+1)  
                          SCPI Compliance:  confirmed

### Description:

The overlay grid (graticule) is not displayed on the analyzer's screen when it is turned off. In addition, the overlay grid does not appear in a plot or print of the screen.



**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:LABel****command/query**

Loads a label for the specified trace.

Command Syntax:           DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:LABel '<STRING>'

<STRING> ::= ASCII characters - 32 through 126  
                  maximum number of characters: 13

Example Statements:        OUTPUT 711;"DISPLAY:WIND3:TRACE:LABEL 'CEPSTRUM' "  
                              OUTPUT 711;"disp:trace:lab 'SPL' "

Query Syntax:             DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:LABel?

Return Format:             "<STRING>"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

Trace titles replace the default titles supplied by the analyzer. They appear above the upper-left corner of the trace box. Trace titles can be a maximum of 13 characters long.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

---

**Note**                    If you send \*RST or SYST:PRES, trace titles are automatically erased. The analyzer restores the default title for all trace boxes.

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:LABel:DEFault[:STATe]****command/query**

Turns on or off the analyzer's default title for the specified trace.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:LABel:DEFault[:STATe]  
                          OFF|0|ON|1

Example Statements:        OUTPUT 711;"disp:wind4:trace:lab:default:stat OFF"  
                          OUTPUT 711;"Disp:Trace:Lab:Default ON"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:TRACe:LABel:DEFault[:STATe]?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:  no  
                          Preset State:  ON (+1)  
                          SCPI Compliance:  instrument-specific

**Description:**

See the DISP[:WINDow[1|2|3|4]]:TRACE:LABEL command for information about providing your own trace titles.

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:X:MATCh[1 | 2 | 3 | 4]****command**

Modifies the X-axis scaling of a trace to match the X-axis scaling of the reference trace.

Command Syntax:           DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:X:MATCh [1 | 2 | 3 | 4]

Example Statements:        OUTPUT 711;"DISPLAY:WIND4:TRAC:X:MATCH3"  
                              OUTPUT 711;"disp:wind2:trac:x:matc1"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   not applicable  
                              SCPI Compliance:   instrument-specific

Description:

This command modifies the scaling of the X-axis of the trace selected with the [:WINDow[1|2|3|4]] trace specifier to match the X-axis scaling of the reference trace. The MATCH[1|2|3|4] trace specifier selects the reference trace. Omit the specifier or use 1 for trace A, 2 for trace B, 3 for trace C, and 4 for trace D.

The analyzer uses the current start and stop X-axis values of the reference trace for the trace selected with the WINDow trace specifier. In addition, the analyzer modifies the spacing of the X-axis (specified with the DISP:TRAC:X:SPAC command) to that of the reference trace.

If the trace formats are not compatible, the X-axis is not modified.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:X[:SCALe]:AUTO****command/query**

Scales the measurement data to fit the trace box.\$Iscaling;automatic

Command Syntax: DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:AUTO OFF|0|ONCE

Example Statements: OUTPUT 711;"DISP:TRAC:X:AUTO ONCE"  
OUTPUT 711;"disp:window3:trac:x:scale:auto once"

Query Syntax: DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:AUTO?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

## Description:

To scale the measurement data, send DISP:TRAC:X:AUTO ONCE.

OFF has no effect on the analyzer. ON is not a valid option, because the analyzer does not support continuous scaling of the data along the X-axis.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:X[:SCALe]:LEFT****command/query**

Specifies the first X-axis value on the display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:LEFT {<number> [ <unit> ] } | <bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV]

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :DISP:TRAC:X:LEFT -8.83941e+37"  
                          OUTPUT 711;"display:wind2:trac:x:scal:left 5.61155e+37"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:LEFT?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.0 HZ (Trace A)  
  +0.0 HZ (Trace B)  
  +0.0 S (Trace C)  
  +0.0 S (Trace D)  
                          SCPI Compliance: confirmed

Description:

This command specifies the value of the first (most left) X-axis point on the display.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

To determine the X-axis unit, send DISP:TRAC:X:LEFT? UNIT.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:X[:SCALe]:RIGHt****command/query**

Specifies the last X-axis value on the display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:RIGHt {<number> [ <unit> ] } | <bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV]

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Disp:Wind:Trace:X:Scale:Righ -9.29266e+37"  
                          OUTPUT 711;"DISP:TRACE:X:RIGHT -1.10689e+37"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X[:SCALe]:RIGHt?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +5.12E+04 HZ (Trace A)  
  +5.12E+04 HZ (Trace B)  
  +7.805E-03 S (Trace C)  
  +7.805-03 S (Trace D)  
                          SCPI Compliance: confirmed

## Description:

This command specifies the value of the last (most right) X-axis point on the display.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

To determine the X-axis unit, send DISP:TRAC:X:RIGH? UNIT.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:X:SPACing****command/query**

Specifies X-axis scaling.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X:SPACing LINear|LOGa-  
rithmic

Example Statements:       OUTPUT 711;":disp:trac:x:spac LINEAR"  
OUTPUT 711;"Display:Wind4:Trac:X:Spac LOGARITHMIC"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:X:SPACing?

Return Format:           LIN|LOG

Attribute Summary:       Option:   not applicable  
Synchronization Required: no  
Preset State:   LIN  
SCPI Compliance: confirmed

## Description:

The trace specifier, WINDOW[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

To select linear scaling of the X-axis, send DISP:TRAC:X:SPAC LIN.

To select logarithmic scaling the X-axis, send DISP:TRAC:X:SPAC LOG.

**DISPlay[:WINDow[1 | 2 | 3 | 4]:TRACe:Y:MATCh[1 | 2 | 3 | 4]****command**

Modifies the Y-axis scaling of a trace to match the Y-axis scaling of the reference trace.

Command Syntax:           DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:Y:MATCh [1 | 2 | 3 | 4]

Example Statements:        OUTPUT 711;":DISP:WIND2:TRAC:Y:MATCH1"  
                               OUTPUT 711;":display>window4:trac:y:match3"

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

## Description:

This command modifies the scaling of the Y-axis of the trace selected with the [:WINDow[1|2|3|4]] trace specifier to match the Y-axis scaling of the reference trace. The MATCH[1|2|3|4] trace specifier selects the reference trace. Omit the specifier or use 1 for trace A, 2 for trace B, 3 for trace C, and 4 for trace D.

The Y-axis of both traces must be compatible. Scaling of the Y-axis is specified with the CALCulate:FORMat command. The following coordinate systems are compatible:

- Linear magnitude for both traces (CALC:FORM MLIN).
- Linear magnitude data on a logarithmic Y-axis for both traces (CALC:FORM MLIN;DISP:TRAC:Y:SPAC LOG).
- Logarithmic magnitude for both traces (CALC:FORM MLOG).
- Phase (wrapped or unwrapped) for both traces (CALC:FORM PHAS or CALC:FORM UPH).
- Numbers (real or imaginary) for both traces (CALC:FORM REAL or CALC:FORM IMAG).
- Nyquist diagram for both traces (CALC:FORM NYQ).
- Polar plots for both traces (CALC:FORM POL).

If the trace formats are not compatible the Y-axis is not modified.



**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y[:SCALe]:AUTO****command/query**

Scales and repositions the trace vertically to provide the best display of trace data.

Command Syntax:           DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:Y [:SCALe] :AUTO OFF | 0 | ON | 1 | ONCE

Example Statements:        OUTPUT 711;"DISPLAY:WIND2:TRAC:Y:SCAL:AUTO ONCE"  
                               OUTPUT 711;"disp:trac:y:auto ON"

Query Syntax:             DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:Y [:SCALe] :AUTO?

Return Format:             ON | OFF

Attribute Summary:        Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:   OFF (+0)  
                               SCPI Compliance:  confirmed

## Description:

To initiate autoscaling of the specified trace, send DISP:TRAC:Y:AUTO ONCE. The analyzer's autoscaling algorithm changes the values of DISP:TRAC:Y:REF and DISP:TRAC:Y:PDIV to optimize Y-axis scaling which provides the best display of your data.

ON autoscales the specified trace after every display update. OFF disables autoscaling.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

**DISPlay[:WINDow[1 | 2 | 3 | 4]:TRACe:Y[:SCALE]:BOTTom****command/query**

Specifies the value of the bottom reference point of the display's Y-axis scale.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALE]:BOTTom {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:9.9e37

<unit> ::= [DBVRMS|VRMS|VPK|DBVPK|V|DBV|EU|DBEU]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :Display:Trac:Y:Bottom 7.35957e+37"  
                          OUTPUT 711;"DISP:WINDOW3:TRAC:Y:SCALE:BOTT 3.87223e+37"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALE]:BOTTom?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

## Description:

This command defines the bottom of a display's Y-axis scale. Specifying a Y-axis per-division value (DISP:TRAC:Y:PDIV) after using this command changes the top and center points of the display. The bottom point remains fixed. When you send this command, the analyzer automatically sets DISP:WIND:TRAC:Y:REF to BOTT.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

See "Determining Units" in appendix E for information about available Y-axis units.

**DISPlay[:WINDow[1 | 2 | 3 | 4]:TRACe:Y[:SCALe]:CENTer****command/query**

Specifies the value of the center reference point of the display's Y-axis scale.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]:TRACe:Y[:SCALe]:CENTer {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [DBVRMS|VRMS|VPK|DBVPK|V|DBV|EU|DBEU]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Display:Wind3:Trac:Y:Scal:Center 0.0"  
                          OUTPUT 711;"DISP:TRAC:Y:CENT -40"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]:TRACe:Y[:SCALe]:CENTer?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

## Description:

This command defines the center of a display's Y-axis scale. Specifying a Y-axis per-division value (DISP:TRAC:Y:PDIV) after using this command changes the top and bottom points of the display. The center point remains fixed. When you send this command, the analyzer automatically sets DISP:WIND:TRAC:Y:REF to CENT.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

See "Determining Units" in appendix E for information about available Y-axis units.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y[:SCALe]:PDIVision****command/query**

Defines the height of each vertical division on the specified trace.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALe]:PDIVision  
                              {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                              limits: 0:9.9e37

<unit> ::= [DB|VRMS|VPK|V]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"display:wind:trac:y:scal:pdivision  
                              9.20888e+37"  
                              OUTPUT 711;"Disp:Trac:Y:Pdiv 4.5054e+37"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALe]:PDIVision?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +10.00 (Trace A and Trace C)  
                              SCPI Compliance: instrument-specific

## Description:

This command compresses or expands displayed data along the Y-axis. The value specifies the height of each vertical division. When trace coordinates are log magnitude (CALC:FORM MLIN;DISP:WIND:TRAC:Y:SPAC LOG), the value specifies the number of displayed decades.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

The preset value for trace B and trace D is determined by the input range.

When trace coordinates are dB magnitude (CALC:FORM MLOG) the only valid unit selection is dB. See "Determining Units" in appendix E for information about available Y-axis units.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y[:SCALe]:REFerence****command/query**

Determines the Y-axis reference position for the specified display.

Command Syntax:           DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:Y [:SCALe] :REFerence  
TOP | CENTer | BOTTOm | RANGe

Example Statements:       OUTPUT 711;"DISP:WIND3:TRAC:Y:REF CENT"  
OUTPUT 711;"display:trace:y:reference bottom"

Query Syntax:             DISPlay [:WINDow [1 | 2 | 3 | 4]] :TRACe:Y [:SCALe] :REFerence?

Return Format:            TOP | CENT | BOTT | RANG

Attribute Summary:       Option:   not applicable  
Synchronization Required: no  
Preset State:   RANGe  
SCPI Compliance: instrument-specific

## Description:

When you change the height of the vertical division of the trace (DISP:TRAC:Y:PDIV) the specified Y-axis reference position remains constant or fixed. The specified trace moves up or down in the display area when you change the reference position.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

- To fix the top of the specified display as the Y-axis reference point, send DISP:TRAC:Y:REF TOP.
- To fix the center of the specified display as the Y-axis reference point, send DISP:TRAC:Y:REF CENT.
- To fix the bottom of the specified display as the Y-axis reference point, send DISP:TRAC:Y:REF BOTT.

To specify a value for the reference point, send the DISP:WIND:TRACE:Y:TOP, DISP:WIND:TRACE:Y:CENT, or DISP:WIND:TRACE:Y:BOTT command.

To select Y-axis scaling which is based on the input range of the channel supplying measurement data, send DISP:TRAC:Y:REF RANG. This is called automatic reference level tracking.

## DISPlay

Your selection of measurement data (CALC:FEED) and trace coordinates (CALC:FORM) affects reference level tracking.

- When linear magnitude trace coordinates are selected, the bottom reference is kept at 0 (zero). The height of the vertical division of the trace (DISP:TRAC:Y:PDIV) is changed so the top reference is the input range.
- When logarithmic magnitude trace coordinates are selected, the top reference is kept at the input range.
- When the real or imaginary trace coordinates are selected, the center reference is set to 0 (zero). The height of the vertical division of the trace (DISP:TRAC:Y:PDIV) is changed so the top reference is the input range.

Reference level tracking is not allowed for the following:

- phase trace coordinates (CALC:FORM:PHAS and CALC:FORM:UPH)
- frequency response measurement data (CALC:FEED 'XFR:POW:RAT 2,1')
- coherence measurement data (CALC:FEED 'XFR:POW:COH 1,2')
- user math data (CALC:MATH:STAT ON)

Reference level tracking is disabled when autoscaling is turned on (DISP:TRAC:X:AUTO or DISP:TRAC:Y:AUTO). It is also disabled when the height of the vertical division (DISP:TRAC:Y:PDIV) changes for real (CALC:FORM REAL), imaginary (CALC:FORM IMAG) or linear magnitude (CALC:FORM MLIN) trace coordinates.

**DISPlay[:WINDow[1 | 2 | 3 | 4]:TRACe:Y[:SCALe]:TOP****command/query**

Specifies the value of the top reference point of the display's Y-axis scale.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALe]:TOP {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [DBVRMS|VRMS|VPK|DBVPK|V|DBV|EU|DBEU]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"DISPLAY:WIND4:TRAC:Y:SCAL:TOP 5.0"  
                          OUTPUT 711;"disp:trac:y:top 0.0"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALe]:TOP?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

## Description:

This command defines the top of a display's Y-axis scale. Specifying the height of the vertical division of the trace (DISP:TRAC:Y:PDIV) after using this command changes the center and bottom points of the display. The top point remains fixed. When you send this command, the analyzer automatically sets DISP:WIND:TRAC:Y:REF to TOP.

The trace specifier, WINDow [1|2|3|4], determines which trace you are selecting. Omit the specifier or send WIND1 for trace A, WIND2 for B, WIND3 for C, or WIND4 for D.

See "Determining Units" in appendix E for information about available Y-axis units.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y:SPACing****command/query**

Specifies scaling of the Y-axis for linear magnitude coordinate data.

Command Syntax:           DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y:SPACing LINear|LOGa-  
rithmic

Example Statements:        OUTPUT 711;":Display:Trac:Y:Spacing Logarithmic"  
OUTPUT 711;":DISP:WIND3:TRAC:Y:SPAC LIN"

Query Syntax:             DISPlay[:WINDow[1 | 2 | 3 | 4]]:TRACe:Y:SPACing?

Return Format:            LIN|LOG

Attribute Summary:        Option:   not applicable  
Synchronization Required: no  
Preset State:   LIN  
SCPI Compliance:   confirmed

**Description:**

The trace specifier, WINDOW[1|2|3|4], determines which trace you are selecting. Omit the specifier or send 1 for trace A, 2 for trace B, 3 for trace C, or 4 for trace D.

To display linear magnitude coordinate data on a linear Y- axis scale, send DISP:TRAC:Y:SPAC LIN.

To display linear magnitude coordinate data on a logarithmic Y-axis scale, send DISP:TRAC:Y:SPAC LOG.

Use this command with the CALC:FORM MLIN command to display linear magnitude data on a logarithmic Y-axis scale, CALC:FORM;MLIN;:DISP:TRAC:Y:SPAC LOG.

---

**Note**                    Only magnitude data can be displayed on a logarithmic Y-axis scale.

---



**DISPlay[:WINDow[1 | 2 | 3 | 4]:WATerfall:BASeline****command/query**

Specifies the percentage of each trace that is concealed in the waterfall display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:BASeline {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:100

<unit> ::= [PCT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"disp:wat:bas 30"  
                          OUTPUT 711;"display:wind3:waterfall:baseline 33"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:BASeline?

Return Format:           Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0  
                          SCPI Compliance: instrument-specific

## Description:

This command allows you to mask a portion of each trace from the waterfall display. The percentage you specify is applied from the baseline of the trace, up towards the peak value and simplifies the waterfall by removing noise floor from the display.

For example, if DISP:WAT:BASE is 33, the lower third of the trace—from the baseline to 33 percent of the amplitude of the trace height—is suppressed. See the DISPlay:WATerfall:HEIGht command for more information.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:WATerfall:BOTTom****command/query**

Specifies the bottom Z-axis value in a waterfall display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:BOTTom {<number>  
                              [<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                              limits: 0:9.9e37

<unit> ::= [S|RPM|COUNT|AVG]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :DISPLAY:WAT:BOTT 3.94581e+37"  
                              OUTPUT 711;"display:wind2:waterfall:bott 8.84371e+37"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:BOTTom?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

## Description:

This command specifies the starting Z-axis value which appears at the bottom of a waterfall display. Use the DISP[:WIND]:WAT:TOP command to specify the corresponding Z-axis value which appears at the top of the waterfall display.

Use the DISP[:WIND]:WAT:COUNT command to specify a range of Z-axis values. The DISP:WAT:COUNT command determines the number of traces displayed in a waterfall. If a range of values is specified, the analyzer holds the DISP:WAT:BOTTom value constant and adjusts the top value.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace-box specifier, the command defaults to trace box A.

**DISPlay[:WINDow[1 | 2 | 3 | 4]:WATerfall:COUNT****command/query**

Determines the number of traces displayed in the waterfall trace box.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:COUNT {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 1e-6:9.9e37

<unit> ::= [S|RPM|COUNT|AVG]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;" :Display:Wat:Coun 5"  
                          OUTPUT 711;"DISP:WIND2:WAT:COUNT 100 RPM"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:WATerfall:COUNT?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +14 COUNT  
                          SCPI Compliance: instrument-specific

## Description:

The number of traces displayed in a waterfall is determined by the range of Z-axis values specified with this command. The analyzer may adjust the specified range to include the trace selected with the CALC:WAT:TRAC:SEL command.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

---

**Note**                    All traces in a waterfall display are deleted when the ABORT;:INIT:CONT ON command is sent.

---

**DISPlay[:WINDow[1 | 2 | 3 | 4]:WATERfall:HEIGHT****command/query**

Specifies the height of a waterfall trace box.

Command Syntax:       DISPlay[:WINDow[1|2|3|4]]:WATERfall:HEIGHT {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 1:100

<unit> ::= [PCT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Disp:Window4:Wat:Height 69"  
                              OUTPUT 711;"DISP:WAT:HEIGHT 84"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATERfall:HEIGHT?

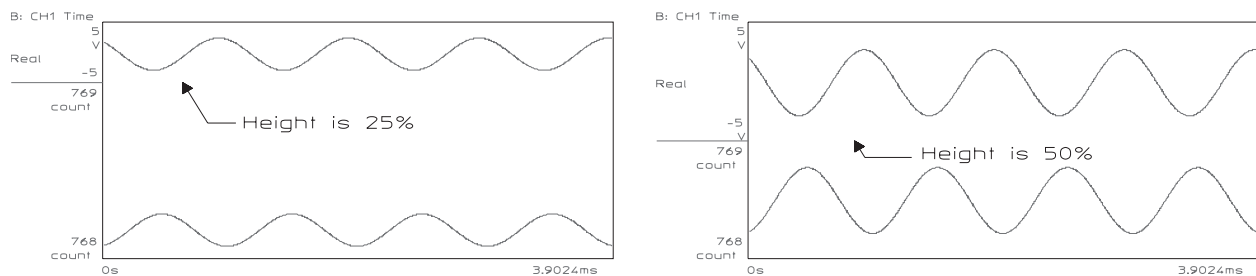
Return Format:           Integer

Attribute Summary:       Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +57 PCT  
                              SCPI Compliance: instrument-specific

Description:

The value you specify determines the height of the waterfall trace box as a percentage of the total height of the waterfall display area. See figure 8-2.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.



**Figure 8-2. Height of the Waterfall Trace Box  
as a Percentage of the Total Height  
of the Waterfall Display Area**

**DISPlay[:WINDow[1 | 2 | 3 | 4]:WATerfall:HIDDen****command/query**

Turns on or off the removal of hidden waterfall traces.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:HIDDen OFF|0|ON|1

Example Statements:       OUTPUT 711;":disp:waterfall:hidd ON"  
                               OUTPUT 711;"Disp:Window4:Wat:Hidden ON"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:WATerfall:HIDDen?

Return Format:             Integer

Attribute Summary:       Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:   ON (+1)  
                               SCPI Compliance:  instrument-specific

**Description:**

In a waterfall display, trace segments may overlap and make it difficult to read the display. When DISP:WAT:HID is ON (the default), the analyzer removes hidden lines—those segments of a trace which fall behind or below the previous trace.

When DISP:WAT:HID is OFF, the analyzer displays all segments of all traces; even those obscured by the previous trace.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:WATerfall:SKEW****command/query**

Enables a skewed waterfall display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:SKEW OFF|0|ON|1

Example Statements:       OUTPUT 711;"display:wind3:waterfall:skew on"  
                          OUTPUT 711;"DISP:WAT:SKEW OFF"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:WATerfall:SKEW?

Return Format:            Integer

Attribute Summary:       Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   OFF (+0)  
                          SCPI Compliance:   instrument-specific

**Description:**

Send DISPlay:WATerfall:SKEW ON to enable a skewed waterfall display. Each trace added to the display is offset along the horizontal axis as well as the vertical axis. The amount of horizontal offset is determined by the DISPlay:WATerfall:SKEW:ANGLE command.

Send DISPlay:WATerfall:SKEW OFF to enable a vertical waterfall display. The offset is only on the vertical axis. This is the default waterfall display.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

**DISPlay[:WINDow[1 | 2 | 3 | 4]]:WATERfall:SKEW:ANGLE****command/query**

Specifies the amount of horizontal offset for waterfall displays.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATERfall:SKEW:ANGLE <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -45:45

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"disp:wat:skew:angl 30"  
                              OUTPUT 711;"DISP:WIND3:WAT:SKEW:ANGLE 20"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:WATERfall:SKEW:ANGLE?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: 30°  
                              SCPI Compliance: instrument-specific

**Description:**

This command is used with the DISPlay:WATERfall:SKEW command. In a waterfall display, a trace is offset from the horizontal axis by the “angle of skew” specified with this command. The trace scrolls down the display towards the origin. See figure 8-3.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

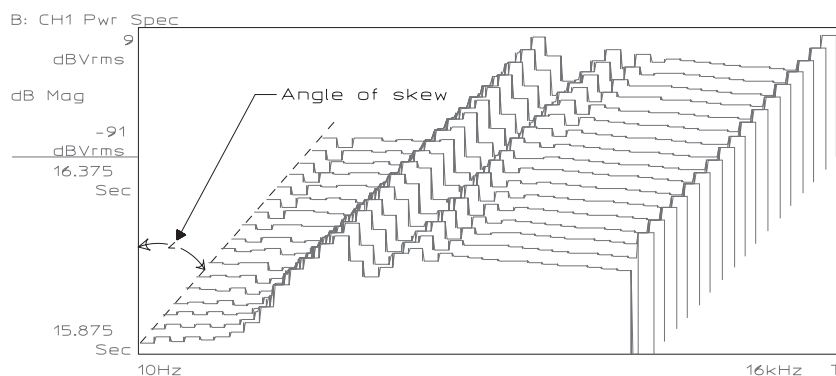


Figure 8-3. Waterfall Traces Displayed  
at 45° Angle of Skew

## DISPlay

---

### **DISPlay[:WINDow[1 | 2 | 3 | 4]:WATerfall[:STATe]**

**command/query**

Turns on or off the waterfall display for the specified trace box.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall[:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;"DISP:WIND:WATERFALL:STAT ON"  
                              OUTPUT 711;"display:wat ON"

Query Syntax:             DISPlay[:WINDow[1|2|3|4]]:WATerfall[:STATe]?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: OFF (+0)  
                              SCPI Compliance: instrument-specific

Description:

To turn on a waterfall display, send DISP:WAT ON.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace box specifier, the command defaults to trace box A.

---

**Note**                    Waterfall displays are not allowed in swept sine instrument mode.

---



**DISPlay[:WINDow[1 | 2 | 3 | 4]]:WATerfall:TOP****command/query**

Specifies the top Z-axis value in a waterfall display.

Command Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:TOP {<num-  
ber>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:9.9e37

<unit> ::= [S|RPM|COUNT|AVG]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :Disp:Waterfall:Top 2.70387e+37"  
                          OUTPUT 711;" DISPLAY:WIND3:WAT:TOP 9.26034e+37"

Query Syntax:           DISPlay[:WINDow[1|2|3|4]]:WATerfall:TOP?

Return Format:           Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

## Description:

This command specifies the starting Z-axis value which appears at the top of a waterfall display. Use the DISP[:WIND]:WAT:BOTTom command to specify the corresponding Z-axis value which appears at the bottom of the waterfall display.

Use the DISP[:WIND]:WAT:COUNT command to specify a range of Z-axis values. The DISP:WAT:COUNT command determines the number of traces displayed in a waterfall. If a range of values is specified, the analyzer holds the DISP:WAT:BOTTom value constant and adjusts the top value.

WINDow[1|2|3|4] specifies the trace box. Send WIND1 to specify A, WIND2 to specify B, WIND3 to specify C, and WIND4 to specify D. If you do not send the optional trace-box specifier, the command defaults to trace box A.



FORMat

---



## FORMat

---

This subsystem contains one command—FORMat:DATA. The command determines which data type and data encoding is used when you transfer blocks of numeric data between the Agilent 35670A and a controller.

**FORMat[:DATA]****command/query**

Specifies the data type and date encoding to be used during transfers of a data block.

Command Syntax:           FORMat [:DATA] ASCii|REAL, [<number>|<bound>]

<number> ::= a real number (NRf data)  
                  limits: 3:64

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"FORMAT:DATA ASCii, 8"  
                          OUTPUT 711;"FORM REAL, 64"

Query Syntax:           FORMat [:DATA] ?

Return Format:           ASC, NR1  
                          REAL, NR1

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: ASC 12  
                          SCPI Compliance: confirmed

Description:

FORM:DATA only affects data transfers initiated by the following commands:

- CALC:DATA?
- CALC:CFIT:DATA?
- CALC:LIM:LOW:REP?
- CALC:LIM:LOW:SEGM
- CALC:LIM:UPP:REP?
- CALC:LIM:UPP:SEGM
- CALC:SYNT:DATA?
- CALC:WAT:DATA?
- PROG:SEL:NUMB
- TRAC[:DATA]
- TRAC:WAT[:DATA]
- TRAC:X[:DATA]?
- TRAC:Z[:DATA]?

FORM:DATA ASC selects extended numeric data for transfers to the analyzer and floating-point-number data for transfers from the analyzer. Data encoding is ASCII. You control the number of significant digits in the returned numbers with the second parameter, which has a range of 3 through 12 when the first parameter is ASC.

---

**Note**                   Data can be sent to the analyzer in ASCii, even if FORMat[:DATA] is REAL.

---

FORM:DATA REAL selects definite or indefinite length block data for transfers to the analyzer but only definite length block data for transfers from the analyzer. Data encoding is binary (the binary floating-point format defined in the IEEE 754-1985 standard). The only allowed values for the second parameter are 32 and 64; it determines how many bits are used for each number.

---

**Note**            It is easiest for Instrument BASIC to read numbers if the format is REAL, 64.

---

See “Data Formats” in the GPIB Programmer’s Guide for more information.





HCOPy

---



## HCOPY

---

The commands in this subsystem control the Agilent 35670A's print and plot operations. It contains commands that allow you to plot different portions of the analyzer's screen, use a time stamp, direct the print/plot operation to the internal disk drive, to the serial port, to the parallel port, or over the GPIB.

---

**HCOPY:COLor:DEFault**

**command**

Specifies default values for the plotter pen assignments.

Command Syntax:            HCOpy:COLor:DEFault

Example Statements:        OUTPUT 711;"hcop:color:def"  
                               OUTPUT 711;"Hcop:Color:Def"

Attribute Summary:        Option: not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

Description:

The default plotter pen assignments are as follows:

- HCOP:ITEM:LAB:COL = 4
- HCOP:ITEM:TRAC:GRAT:COL = 1
- HCOP:ITEM:WIND1:TRAC:MARK:COL = 2
- HCOP:ITEM:WIND2:TRAC:MARK:COL = 3
- HCOP:ITEM:WIND3:TRAC:MARK:COL = 5
- HCOP:ITEM:WIND4:TRAC:MARK:COL = 6
- HCOP:ITEM:WIND1:TRAC:COL = 2
- HCOP:ITEM:WIND2:TRAC:COL = 3
- HCOP:ITEM:WIND3:TRAC:COL = 5
- HCOP:ITEM:WIND4:TRAC:COL = 6

**HCOPY:DESTination**

**command/query**

Specifies where the print or plot operation is sent: either to a device or to a file on the default disk.

Command Syntax:            HCOpy:DESTination <data\_handler>

```
data_handler ::= `SYSTem:COMMunicate:GPIB:RDEvice' |
              `SYSTem:COMMunicate:GPIB:RDEvice"
              ::= `MMEMory'|"MMEMory"
              ::=
                                     `SYSTem:
                                     COMMunicate:CENTronics'|"SYSTem:COMMunicate:CENTronics"
              ::= `SYSTem:COMMunicate:SERial'|"SYSTem:COMMunicate:SERial"
```

Example Statements:        OUTPUT 711;"HCOP:DEST `MMEM' "  
                               OUTPUT 711;"Hcopy:Destination  
                               `System:Communicate:Gpib:Rdevice' "

Query Syntax:             HCOpy:DESTination?

Return Format:             "MMEM"|"SYST:COMM:GPIB:RDEV"|"SYST:COMM:CENT" |  
                               "SYST:COMM:SER"

Attribute Summary:        Option: not applicable  
                               Synchronization Required: no  
                               Preset State: `SYST:COMM:GPIB:RDEV'  
                               SCPI Compliance: approved

Description:

This command specifies the destination only. Use the HCOP[:IMM] command to initiate the print/plot operation. Use the HCOP:DEV:LANG command to specify the format of the output.

‘SYSTem:COMMunicate:GPIB:RDEvice’ sends the print/plot directly to the GPIB device. Use the HCOpy:PRIN:ADDR or HCOP:PLOT:ADDR command to assign an GPIB address to your device.

‘MMEMory’ sends the print/plot to a file on the default disk. Specify the filename with the MMEM:NAME command. For information on how to send the print/plot operation to a file on a mass storage device other than the default disk, see the MMEMory:NAME command.

‘SYSTem:COMMunicate:PARallel’ sends the print/plot to the device connected to the parallel (Centronics) port on the rear panel.

‘SYSTem:COMMunicate:SERial’ sends the print/plot to the device connected to the serial (RS-232-C) port on the rear panel. Use the SYSTem:COMMunicate:SERial commands to configure the RS-232-C interface.

---

**HCOPY:DEVIce:LANGuage**

**command/query**

Specifies the format of the plot/print output.

Command Syntax:           HCOPY:DEVIce:LANGuage HPGL|PCL|PHPGL

Example Statements:        OUTPUT 711;":hcopy:device:language hppl"  
                               OUTPUT 711;"HCOP:DEV:LANG PCL"

Query Syntax:             HCOPY:DEVIce:LANGuage?

Return Format:             HPGL|PCL|PHPGL

Attribute Summary:        Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:  HPGL  
                               SCPI Compliance:  approved, except PHPGL

Description:

Use HPGL to specify the format for a plotter.

Use PCL to specify the format for a printer in raster mode. Use PHPGL to specify the format for a printer that understands HPGL.

---

**Note**                    When PCL is selected, the entire screen is always printed.

---

**HCOpy:DEvice:RESolution****command/query**

Specify the format of data saved to a file.

Command Syntax:           HCOpy:DEvice:RESolution <number>

                  <number> ::= a real number (NRf data)  
                                  limits: 0 to 32767

Example Statements:       OUTPUT 711;"HCOP:DEV:RES 75"

Query Syntax:             HCOpy:DEvice:RESolution?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset state: 0  
                          SCPI Compliance: instrument-specific

Description:

Use this command to specify the printer resolution in dots per inch for raster printer plots.

Changing the printer resolution will change the size of the output plot. A printer resolution of 75 will give a full page plot.

Set the resolution to zero if you want to use the printer's default resolution.

**HCOPY:DEVIce:SPEed****command/query**

Specifies the plotting speed for all plotting operations initiated by the analyzer.

Command Syntax:           HCOPY:DEVIce:SPEed <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:100

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"HCOP:DEV:SPE 53"  
                          OUTPUT 711;"hcopy:device:speed 0"

Query Syntax:             HCOPY:DEVIce:SPEed?

Return Format:            Integer

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0 (plotter's default speed)  
                          SCPI Compliance: approved

**Description:**

This command allows you to specify the plotting speed in units of centimeters per second (cm/s). Check your plotter's documentation to be sure that it supports the requested plotting speed.

For example, send HCOP:DEV:SPE 75 to select a plotting speed of 75 cm/second. Send HCOP:DEV:SPE 10 to select a slower plotting speed of 10 cm/second. HCOP:DEV:SPE 0 selects the plotter's default plot speed.



---

**HCOpy[:IMMEDIATE]****command**

Plots or prints the currently specified item.

Command Syntax:           HCOpy[:IMMEDIATE]

Example Statements:        OUTPUT 711;"HCOpy"  
                              OUTPUT 711;"hcop:imm"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   approved

Description:

This command prints or plots only the item(s) specified with the HCOP:SOUR command. This command is comparable to the front-panel [START PLOT/PRNT] softkey. Use the HCOP:DEST command to specify where the print/plot operation is sent. Use the HCOP:DEV:LANG to specify the format of the output.

---

**Note**            Pass Control Required if HCOP:DEST is 'SYST:COMM:GPIB:RDEV' (GPIB).  
                  See "Passing Control" in chapter 3 of the *GPIB Programmer's Guide* for  
                  information on how to pass control to the analyzer.

---

---

**HCOPY:ITEM:FFEed:STATe**

**command/query**

Turns the page-eject feature on or off.

Command Syntax:            HCOpy:ITEM:FFEed:STATe OFF|0|ON|1

Example Statements:        OUTPUT 711;":hcopy:item:ffe:state ON"  
                               OUTPUT 711;"Hcop:Item:Ffe:Stat OFF"

Query Syntax:              HCOpy:ITEM:FFEed:STATe?

Return Format:              Integer

Attribute Summary:        Option:    not applicable  
                               Synchronization Required: no  
                               Preset State:    ON (+1)  
                               SCPI Compliance: approved

Description:

The page-eject occurs after the plot/print is completed.

Check the documentation for your device to verify that it supports the requested page-eject state.

**HCOpy:ITEM:LABel:COLor****command/query**

Selects the pen used for plotting miscellaneous annotations.

Command Syntax:           HCOpy:ITEM:LABel:COLor <number>|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 0:16
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements:       OUTPUT 711;"hcop:item:lab:col 4"  
                          OUTPUT 711;" :HCOpy:ITEM:LABel:COLOR 5"

Query Syntax:            HCOpy:ITEM:LABel:COLor?

Return Format:           Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +4  
                          SCPI Compliance: instrument-specific

Description:

The label pen is used to plot the following:

- Instrument state
- Disk catalog
- Print and plot output labels
- Mini-state
- Fault log
- Test log
- Time stamp

Nothing is plotted with a pen whose value is specified as 0 or with a pen whose specified value is too large for your plotter.

**HCOPY:ITEM:LABel:STATe**

**command/query**

Prints a label on the plot and print output. abeling:plot/print

Command Syntax: HCOpy:ITEM:LABel:STATe OFF|0|ON|1

Example Statements: OUTPUT 711;"HCOpy:ITEM:LABel:STAT OFF"  
OUTPUT 711;"hcop:item:lab:state ON"

Query Syntax: HCOpy:ITEM:LABel:STATe?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF (+0)  
SCPI Compliance: approved

Description:

Send HCOpy:ITEM:LAB:STAT ON to name your plot or print output. The label appears at the top of the screen. See figure 10-1. The label is plotted (or printed) when the HCOpy:IMM command is executed.

To specify the label, use the HCOpy:ITEM:LABel:TEXT command.

The color of the label is set with the HCOpy:ITEM:LABel:COLOR command.

Send HCOpy:ITEM:LAB:STAT OFF to have the mini-state appear on your plot or print output.

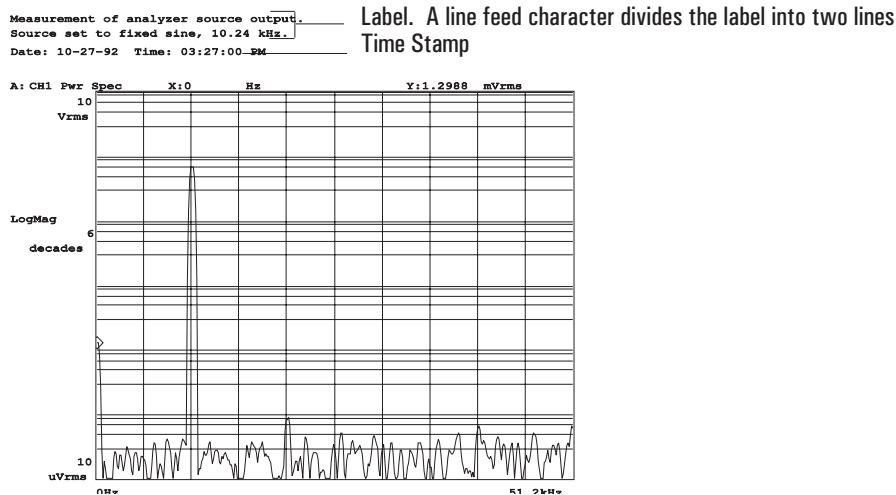


Figure 10-1. Plot/Print Label and Time Stamp

**HCOPY:ITEM:LABel:TEXT****command/query**

Specifies a label for plot and print output. ~~abeling;plot/print~~

Command Syntax:           HCOPY:ITEM:LABel:TEXT '<STRING>'

<STRING> ::= maximum of 119 ASCII characters  
          59 characters per line

Example Statements:       OUTPUT 711;":hcopy:item:lab:text 'Test 1 Results'  
                          OUTPUT 711;":HCOP:ITEM:LABEL:TEXT "LINE1"&CHR\$(10)&"LINE2"

Query Syntax:             HCOPY:ITEM:LABel:TEXT?

Return Format:            "<STRING>"

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: approved

## Description:

This command specifies a label for your plot or print output. The label appears at the top of the screen. See figure 10-1 on page 10-25. The label is plotted (or printed) when HCOP:ITEM:LAB:STAT is ON and the HCOP:IMM command is executed.

A line feed character (decimal value 10) in <STRING>, divides the label into two lines.

Specify a label with a space to have the area appear blank on the plot (the mini-state will not appear).

If you can use a command that does not comply with SCPI, see the HCOP:TITLe command.

**HCOPY:ITEM:TDSTamp:FORMat**

**command/query**

Specifies the format of the time stamp used for plotting and printing.

Command Syntax:	HCOPY:ITEM:TDSTamp:FORMat FORMat1   FORMat2   FORMat3   FORMat4   FORMat5
Example Statements:	OUTPUT 711;":Hcop:Item:Tdstamp:Form FORMAT1" OUTPUT 711;":HCOPY:ITEM:TDST:FORMAT FORMAT2"
Query Syntax:	HCOPY:ITEM:TDSTamp:FORMat?
Return Format:	FORM1   FORM2   FORM3   FORM4   FORM5
Attribute Summary:	Option: not applicable Synchronization Required: no Preset State: FORM5 SCPI Compliance: instrument-specific

Description:

This command selects the time and date format for plot and print operations.

- To select a 24 hour, Day/Month/Year Hour:Minute:Second format, send HCOPY:ITEM:TDSTAMP:FORMAT FORM1.
- To select a 24 hour, Day. Month.Year Hour:Minute:Second format, send HCOPY:ITEM:TDSTAMP:FORMAT FORM2.
- To select a 24 hour, Year Month Day Hour:Minute:Second format, send HCOPY:ITEM:TDSTAMP:FORMAT FORM3.
- To select a 12 hour, Day/Month/Year Hour:Minute:Second AM format, send HCOPY:ITEM:TDSTAMP:FORMAT FORM4.
- To select a 12 hour, Month-Day-Year Hour:Minute:Second AM format, send HCOPY:ITEM:TDSTAMP:FORMAT FORM5.

**HCOPY:ITEM:TDSTamp:STATe****command/query**

Turns a time stamp on or off for print and plot operations.

Command Syntax:           HCOPY:ITEM:TDSTamp:STATe OFF|0|ON|1

Example Statements:       OUTPUT 711;"hcop:item:tdst:stat on"  
                          OUTPUT 711;"HCOPY:ITEM:TDSTAMP:STAT OFF"

Query Syntax:            HCOPY:ITEM:TDSTamp:STATe?

Return Format:            Integer

Attribute Summary:       Option:   not applicable  
                          Synchronization Required:  no  
                          Preset State:  ON (+1)  
                          SCPI Compliance:  approved

**Description:**

When time stamp is ON, time and date information is printed with the screen data you specify with the HCOP:SOUR command. Use the HCOP:ITEM:TDST:FORM command to specify the time stamp format. Figure 10-1 (page 10-1010) illustrates the use of a time stamp (FORMat 2).

When time stamp is OFF, time and date information is not printed.

**HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:COLor**

**command/query**

Selects the pen used to plot the specified trace and annotation.

Command Syntax:            HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:COLor <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:16

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"HCOP:ITEM:WIND3:TRAC:COL 0"  
                          OUTPUT 711;":hcopy:item:window:trace:color 6"

Query Syntax:             HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:COLor?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +2 (TRACeA)  
   +3 (TRACeB)  
   +5 (TRACeC)  
   +6 (TRACeD)  
                          SCPI Compliance: approved

Description:

The trace pen is used to plot traces and all of the following trace-specific annotation:

- Trace title
- Marker readout
- X-axis annotation
- Y-axis annotation

The trace specifier you send with this command determines which trace you are selecting. Omit the specifier or send 1 for trace A; send 2 for trace B; send 3 for trace C; send 4 for trace D.

Nothing is plotted with a pen whose value is specified as 0 or with a pen whose specified value is too large for your plotter.



**HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATicule:COLor****command/query**

Selects the pen used to plot the trace graticules.

Command Syntax:           HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATicule:COLor  
                          <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                          limits: 0:16

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":hcopy:item:trace:graticule:color 8"  
                          OUTPUT 711;":HCOPY:ITEM:TRAC:GRAT:COL 15"

Query Syntax:             HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:GRATicule:COLor?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +1  
                          SCPI Compliance: approved

**Description:**

The trace graticule pen is used to plot the overlay grid, the border around the instrument state and the border around the disk catalog.

Nothing is plotted with a pen whose value is specified as 0 or with a pen whose specified value is too large for your plotter.

---

**Note**                    This command is not trace specific. It ignores the trace specifier.

---

**HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LIMit:LTYPe**

**command/query**

Selects the line type for the specified limit line.

Command Syntax:           HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LIMit  
                              SOLid|DASHed|DOTTed|STYLE<n>

<n> ::= a real number  
          limits: 0, 3-12

Example Statements:        OUTPUT 711;"HCOP:ITEM:TRAC:LIM:LTYP DASH"  
                              OUTPUT 711;"hcopy:item>window4:trace:limit:ltype dotted"

Query Syntax:             HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LIMit:LTYPe?

Return Format:             SOL|DASH|DOTT|STYL<n>

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: DOTT (all traces)  
                              SCPI Compliance: instrument-specific

Description:

The limit specifier determines which trace-limit line you are selecting. Omit the specifier or send 1 for trace A limit lines; send 2 for trace B limit lines; send 3 for trace C limit lines; send 4 for trace D limit lines.

The color of the limit line is set with the HCOP:ITEM:WIND:TRAC:MARK:COL command.

The STYLE<n> parameter values choose a line type <n>. Check your plotter's documentation to see if it supports additional line types. The numerical representation of the line type can be appended to STYLE to select that line type.

**HCOpy:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LTYPe****command/query**

Selects the line type for the specified trace.

Command Syntax:           HCOpy:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LTYPe  
                          SOLid|DASHed|DOTTed|STYLE<n>

<n> ::= a real number  
          limits: 0, 3-12

Example Statements:       OUTPUT 711;"HCOP:ITEM:TRAC:LTYP DASH"  
                          |OUTPUT 711;"hcopy:item>window4:trace:ltype solid"

Query Syntax:            HCOpy:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:LTYPe?

Return Format:            SOL|DASH|DOTT|STYL<n>

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: SOL (all traces)  
                          SCPI Compliance: approved

**Description:**

The trace specifier determines which trace you are selecting. Omit the specifier or send 1 for trace A; send 2 for trace B; send 3 for trace C; send 4 for trace D.

The STYLE<n> parameter values choose a line type <n>. Check your plotter's documentation to see if it supports additional line types. The numerical representation of the line type can be appended to STYLE to select that line type.

**HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:MARKer:COLor**

**command/query**

Selects the pen used to plot markers for the specified trace.

Command Syntax:           HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:MARKer:COLor <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:16

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"hcopy:item>window:trace:marker:color 4"  
                              OUTPUT 711;"HCOP:ITEM:WIND3:TRAC:MARK:COL 9"

Query Syntax:             HCOPY:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:MARKer:COLor?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +2 (MARKA)  
  +3 (MARKB)  
  +5 (MARKC)  
  +6 (MARKD)  
                              SCPI Compliance: instrument-specific

Description:

The marker pen is used to plot all markers; including the main markers, limit lines, the marker reference and marker functions.

The trace specifier you send with this command determines which trace marker you are selecting. Omit the specifier or send 1 for trace A markers; send 2 for trace B markers; send 3 for trace C markers; send 4 for trace D markers.

Nothing is plotted with a pen whose value is specified as 0 or with a pen whose specified value is too large for your plotter.

**HCOPY:PAGE:DIMensions:AUTO****command/query**

Specifies P1 and P2 values for a plotter.

Command Syntax:           HCOPY:PAGE:DIMensions:AUTO OFF|0|ON|1

Example Statements:        OUTPUT 711;"HCOP:PAGE:DIM:AUTO ON"  
                              OUTPUT 711;"hcopy:page:dimensions:auto OFF"

Query Syntax:             HCOPY:PAGE:DIMensions:AUTO?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: approved

**Description:**

Send HCOP:PAGE:DIM:AUTO ON if you wish to use the plotter's current P1 and P2 settings. The analyzer does not send P1 and P2 values to the plotter.

Send HCOP:PAGE:DIM:AUTO OFF if you want to set your own P1 and P2 values. The analyzer sends commands to the plotter which programs the P1 and P2 values when the HCOP:IMM command is executed. P1 and P2 values are specified with the HCOP:PAGE:DIM:LLEF and HCOP:PAGE:DIM:URIG commands.

**HCOPY:PAGE:DIMensions:USER:LLEFt**

**command/query**

Specifies the lower left position (P1) of the plot area.

**Command Syntax:**            HCOPY:PAGE:DIMensions:USER:LLEFt {<P1\_X-axis\_value>,  
                                  <P1\_Y-axis\_value>}

```
<P1_X-axis_value> ::= <number>|<step>|<bound>
<number> ::= a real number (NRf data)
           limits: -32767:32767
<step> ::= UP|DOWN
<bound> ::= MAX|MIN
<P1_Y-axis_value> ::= <number>|<step>|<bound>
<number> ::= a real number (NRf data)
           limits: -32767:32767
<step> ::= UP|DOWN
<bound> ::= MAX|MIN
```

**Example Statements:**        OUTPUT 711;"HCOP:PAGE:DIM:USER:LLEF 332,4286"  
                                  OUTPUT 711;"hcopy:page:dimensions:user:lleft 4743,1195"

**Query Syntax:**               HCOPY:PAGE:DIMensions:USER:LLEFt?

**Return Format:**               Integer, Integer

**Attribute Summary:**        Option: not applicable  
                                  Synchronization Required: no  
                                  Preset State: not affected by Preset  
                                  SCPI Compliance: instrument-specific

**Description:**

The plot area is defined by the X-axis and Y-axis values for the scaling points P1 and P2. See figure 10-2. The X-axis value and the Y-axis value depends upon the plotter and the number of points on the page.

Use the HCOP:PAGE:DIM:USER:URIG command to specify the P2 X- and Y-axis values.

Check the documentation for your plotter to select appropriate P1 values.

**Note**                    HCOP:PAGE:DIM:USER:LLEF is set to +332,+1195 after a reset (\*RST).

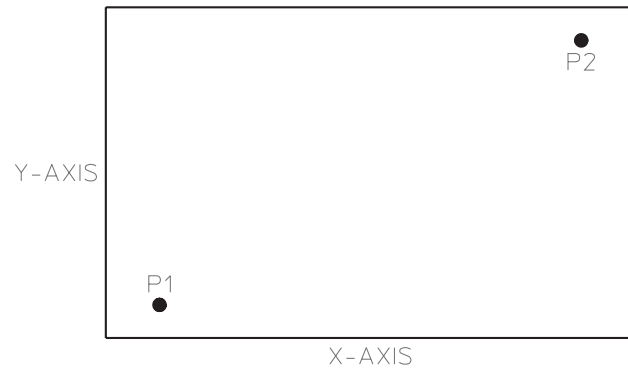


Figure 10-2. Location of Scaling Points P1 and P2

---

**HCOPY:PAGE:DIMensions:USER:URIGHt**

**command/query**

Specifies the upper right position (P2) of the plot area.

**Command Syntax:**            HCOpy:PAGE:DIMensions:USER:URIGHt {<P2\_X-axis\_value>,<P2\_Y-axis\_value>}

```
<P2_X-axis_value> ::= <number>|<step>|<bound>
    <number> ::= a real number (NRf data)
                limits: -32767:32767
    <step> ::= UP|DOWN
    <bound> ::= MAX|MIN
<P2_Y-axis_value> ::= <number>|<step>|<bound>
    <number> ::= a real number (NRf data)
                limits: -32767:32767
    <step> ::= UP|DOWN
    <bound> ::= MAX|MIN
```

**Example Statements:**        OUTPUT 711;"HCOP:PAGE:DIM:USER:URIG 4743,7377"  
                               OUTPUT 711;"hcopy:page:dimensions:user:uright 9155,4286"

**Query Syntax:**               HCOpy:PAGE:DIMensions:USER:URIGHt?

**Return Format:**               Integer, Integer

**Attribute Summary:**        Option: not applicable  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: instrument-specific

**Description:**

The plot area is defined by the X-axis and Y-axis values for the scaling points P1 and P2. The X-axis value and the Y-axis value depends upon the plotter and the number of points on the page.

Use the HCOP:PAGE:DIM:USER:LLEF command to specify the P1 X- and Y-axis values.

Check the documentation for your plotter to select appropriate P2 values.

---

**Note**                        HCOP:PAGE:DIM:USER:URIG is set to +9155,+7377 after a reset (\*RST).

---



**HCOpy:PLOT:ADDRess****command/query**

Tells the analyzer which GPIB address is assigned to your plotter.

Command Syntax:           HCOpy:PLOT:ADDRess <number>|<step>|<bound>

    <number> ::= a real number (NRf data)

                  limits: 0:30

    <step> ::= UP|DOWN

    <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"HCOpy:PLOT:ADDRESS 5"  
                              OUTPUT 711;"hcop:plot:addr 9"

Query Syntax:             HCOpy:PLOT:ADDRess?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: instrument-specific

**Description:**

This command works when the HCOpy:DESTination command is set to GPIB and the HCOpy:DEVIce command is set to PLOT.

Initiate a plot with the HCOpy[:IMM] command. The analyzer expects to find a plotter at the GPIB address specified with HCOpy:PLOT:ADDR. If a plotter is not at the specified address, the plot is automatically aborted.

---

**HCOPY:PRINT:ADDRESS**

**command/query**

Tells the analyzer which GPIB address is assigned to your printer.

Command Syntax:            HCOpy:PRINT:ADDRESS <number>|<step>|<bound>

    <number> ::= a real number (NRf data)  
                   limits: 0:30  
     <step> ::= UP|DOWN  
     <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"HCOPY:PRINT:ADDRESS 3"  
                               OUTPUT 711;"HCOP:PRIN:ADDR 1"

Query Syntax:             HCOpy:PRINT:ADDRESS?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: instrument-specific

**Description:**

This command works when the HCOP:DESTINATION command is set to GPIB and the HCOP:DEVICE command is set to PRINT.

Initiate a print operation with the HCOpy[:IMM] command. The analyzer expects to find a printer at the GPIB address specified with HCOP:PRIN:ADDR. If a printer is not at the specified address, the print operation is automatically aborted.

**HCOPy:TITLe[1 | 2]**

**command/query**

Specifies a label for plot and print output. **abeling:plot/print**

Command Syntax: HCOPy:TITLe [1 | 2 | 3 | 4] '<STRING>'

<STRING> ::= maximum of 59 ASCII characters

Example Statements: OUTPUT 711;":hcopy:titl:lab:text 'Test 1 Results'  
 OUTPUT 711;":HCOP:TITLe:TEXT2 'BEARING CHARACTERISTICS'

Query Syntax: HCOPy:TITLe [1 | 2] ?

Return Format: "<STRING>"

Attribute Summary: Option: not applicable  
 Synchronization Required: no  
 Preset State: not applicable  
 SCPI Compliance: instrument-specific

Description:

This command specifies a label for your plot or print output. The label appears at the top of the screen. The two lines. See figure 10-3. The label is plotted (or printed) when HCOP:ITEM:LAB:STAT is ON and the HCOP:IMM command is executed.

The label specifier determines whether you are specifying text for the first or the second line. Omit the specifier or send 1 for the first line of the label and 2 for the second line of the label.

Specify a label with a space to have the area appear blank on the plot (the mini-state will not appear).

If you need to use a command that complies with SCPI, use the HCOP:ITEM:LABel:TEXT command.

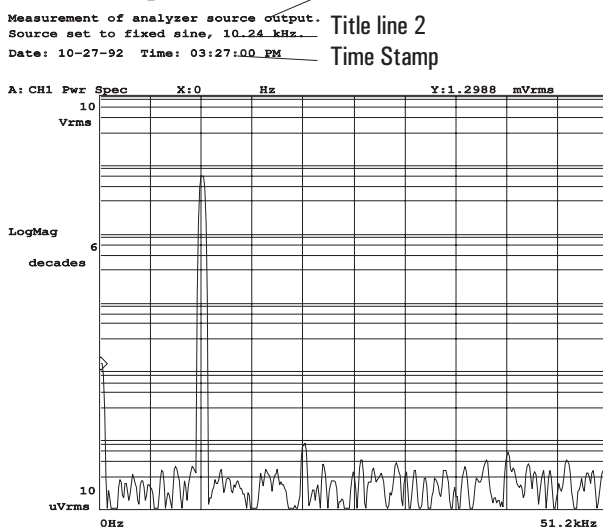


Figure 10-3. Plot/Print Title and Time Stamp

**HCOPY:SOURce**

**command/query**

Selects the portion of the analyzer's screen you want to plot.

Command Syntax:            HCOpy:SOURce {ALL|TRACe|MARKer|REFerence|GRID}

Example Statements:        OUTPUT 711;"HCOP:SOUR GRID"  
                               OUTPUT 711;"hcopy:sour MARKer"

Query Syntax:              HCOpy:SOURce?

Return Format:              CHAR

Attribute Summary:        Option: not applicable  
                               Overlapped: no  
                               Preset State: ALL  
                               SCPI Compliance: instrument-specific

Description:

This command is not valid when the specified device is a printer (HCOP:DEV PRIN). This command can only be used with the HCOP:DEV PLOT command.

To plot everything currently displayed on the analyzer's screen, send HCOP:SOUR ALL;IMM . Everything on the screen is plotted except the status line and the softkey menu.

To plot the displayed trace(s), send HCOP:SOUR TRAC;IMM. Traces are plotted without grid lines, annotation or markers.

Send HCOP:SOUR MARK;IMM to plot the main marker for all displayed trace(s) The main marker must be displayed (CALC:MARK ON) before it can be plotted. The marker is annotated with its X-axis and Y-axis coordinates. The annotation appears above the marker.

Send HCOP:SOUR REF;IMM to plot the marker reference. The marker reference must be displayed (CALC:MARK:MODE REL) before it can be plotted. The marker reference is annotated with its X-axis and Y-axis coordinates. The annotation appears above the marker reference.

Send HCOP:SOUR GRID;IMM to plot the graticule only for all displayed traces. The grid is plotted without the trace, markers or annotation

INITiate

---



# INITiate

---

The INITiate subsystem controls the initiation of the TRIGger system. The commands initiate all TRIGger sequences as a group.

Figure 11-1 shows the model for the Agilent 35670A's ARM-INITiate-TRIGger functions.

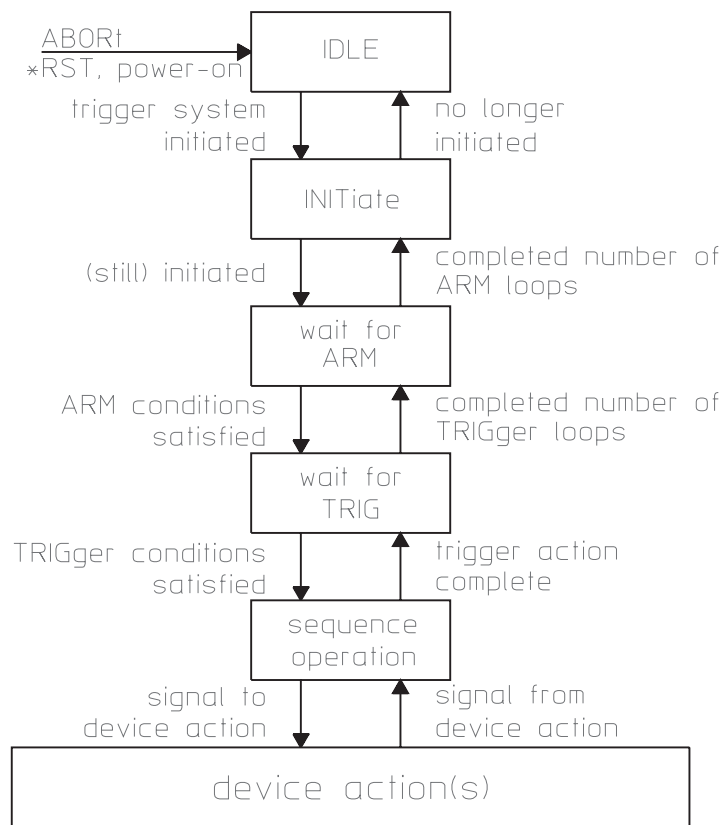


Figure 11-1. The Agilent 35670A's ARM-INIT-TRIG Functions

---

**INITiate:CONTinuous****command/query**

Sets the trigger system to a continuously initiated state.

Command Syntax:           INITiate:CONTinuous OFF|0|ON|1

Example Statements:        OUTPUT 711;"Init:Cont OFF"  
                              OUTPUT 711;"INITIATE:CONT OFF"

Query Syntax:             INITiate:CONTinuous?

Return Format:            Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   no  
                              Preset State:   ON (+1)  
                              SCPI Compliance:   confirmed

Description:

INIT:CONT OFF suspends the measurement process. It “pauses” the current measurement immediately. The analyzer discards the time record that is currently being processed.

INIT:CONT ON restarts a “paused” measurement. It also allows you to add more data to the running average of a completed measurement. For example, if the analyzer has completed a 10-average measurement and you send INIT CONT ON, 10 more records are averaged with the old data, bringing the total number of averages to 20.

INIT:CONT ON is not valid if you pause a measurement and change the measurement parameters. It is also not valid, if a calibration occurs while a measurement is paused. You must use the INIT[:IMM] command to start a paused measurement after calibration or after changing the measurement parameters.

---

**Note**                    After \*RST, INIT:CONT is set to OFF (+0).

---



**INITiate[:IMMEDIATE]****command**

Starts a measurement and forces the trigger system to exit the idle state.

Command Syntax:           INITiate[:IMMEDIATE]

Example Statements:        OUTPUT 711;":initiate"  
                              OUTPUT 711;"Init:Imm"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:   yes  
                              Preset State:   not applicable  
                              SCPI Compliance:   confirmed

**Description:**

This command starts a new measurement and ensures that any changes made to the analyzer's state are reflected in the measurement results. The new measurement is started immediately whether the current measurement is running, "paused," or completed. All data from the previous measurement is discarded when the new measurement is started.

INIT[:IMM] causes the trigger system to initiate and complete one full trigger cycle.

If the command INIT:CONT ON has been sent, the INIT[:IMM] command has no affect.

The program message ABOR;:INIT:IMM serves a special synchronization function. When you send this message to restart a measurement, the analyzer's No Pending Operation (NPO) flag is set to 1 until the measurement is complete. The two commands that test the state of this flag—\*WAI and \*OPC—allow you to hold off subsequent actions until the measurement is complete. See chapter 3 of the GPIB Programmer's Guide for more information about synchronization.



INPut

---



## INPut

---

The commands in this subsystem control the characteristics of the analyzer's input channels. They configure the inputs for Channel 1, Channel 2, Channel 3, and Channel 4.

Because there are two input channels in the Agilent 35670A and four channels in the Agilent 35670A (Option AY6), you must specify the channel you want to configure when you send a command. If you do not explicitly specify one of the channels, the analyzer configures Channel 1.

---

**Note** The Agilent 35670A has up to four input channels (1,2, 3 and 4) and four trace boxes (A, B, C, and D). None of the channels are linked to a particular trace box. You can display data from any input channel in any trace box.

---

**INPut[1 | 2 | 3 | 4]:BIAS[:STATe]****command/query**

Enables/disables the ICP supply on the corresponding input channel.

Command Syntax:           INPut [1 | 2 | 3 | 4] :BIAS [:STATe] OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;"INPUT3:BIAS:STATE ON"  
                          OUTPUT 711;"inp3:bias OFF"

Query Syntax:             INPut [1 | 2 | 3 | 4] :BIAS [:STATe] ?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: OFF (+0)  
                          SCPI Compliance: confirmed

**Description:**

This command connects (or disconnects) the internal 4 mA current source to the input connector. The nominal voltage output is 24 V dc (open circuit).

To get rid of the DC bias that is generated when INP:BIAS is ON, turn on AC coupling. See the INPut:COUPLing command.

If the channel specifier is not used, the command defaults to channel 1.

---

**INPut[1 | 2 | 3 | 4]:COUPling****command/query**

Selects AC or DC coupling for the specified channel.

Command Syntax:           INPut [1 | 2 | 3 | 4] :COUPling AC | DC

Example Statements:       OUTPUT 711;":Input4:Coup AC"  
                          OUTPUT 711;"INPUT:COUP AC"

Query Syntax:             INPut [1 | 2 | 3 | 4] :COUPling?

Return Format:            AC | DC

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: DC  
                          SCPI Compliance: confirmed

Description:

If the channel specifier is not used, the command defaults to channel 1.

**INPut[1 | 2 | 3 | 4]:FILTer:AWEighting[:STATe]****command/query**

Enables/disables the A-weight filter on the specified input channel.

Command Syntax:           INPut [1 | 2 | 3 | 4] :FILTer:AWEighting [:STATe] OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;"inp4:filter:awe:state ON"  
                          OUTPUT 711;"Inp2:Filt:Aweighting OFF"

Query Syntax:             INPut [1 | 2 | 3 | 4] :FILTer:AWEighting [:STATe] ?

Return Format:            Integer

Attribute Summary:       Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   OFF (+0)  
                          SCPI Compliance:   confirmed

Description:

The A-weight filter is normally used with octave measurements.

If the channel specifier is not used, the command defaults to channel 1.



**INPut[1 | 2 | 3 | 4]:FILTer[:LPASs][:STATe]****command/query**

Enables/disables the anti-alias filter for the specified input channel.

Command Syntax:           INPut [1 | 2 | 3 | 4] :FILTer [:LPASs] [:STATe] OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;":INP2:FILTER OFF"  
                               OUTPUT 711;"inp:filt:lpass:stat ON"

Query Syntax:             INPut [1 | 2 | 3 | 4] :FILTer [:LPASs] [:STATe] ?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                               Synchronization Required: no  
                               Preset State: ON (+1)  
                               SCPI Compliance: confirmed

## Description:

When INPut[1|2|3|4]:FILTer is OFF, the analyzer's corresponding input bypasses the anti-alias low pass filter. Measurement results are not corrected for front end flatness; only front end DC offset is calibrated.

If the channel specifier is not used, the command defaults to channel 1.

With the Agilent 35670A Option AY6 (4 input channels), disabling the anti-alias filter in one channel instrument mode (INP2 OFF), allows you to work with a 102.4 kHz frequency span.

---

**Note**                    The anti-alias low pass filter is always bypassed in histogram instrument mode.

---

---

**INPut[1 | 2 | 3 | 4]:LOW****command/query**

Sets the specified channel's input shield to float or to ground.

Command Syntax:           INPut [1 | 2 | 3 | 4] :LOW GROUnd | FLOat

Example Statements:        OUTPUT 711;"Input3:Low GROUND"  
                              OUTPUT 711;"INP2:LOW FLOAT"

Query Syntax:             INPut [1 | 2 | 3 | 4] :LOW?

Return Format:             GRO | FLO

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:   FLO  
                              SCPI Compliance:  confirmed

**Description:**

To connect the analyzer's input shield to ground through 55  $\Omega$ , send INP[1|2|3|4]:LOW GRO.

To float the analyzer's input shield through 1 M $\Omega$ , send INP[1|2|3|4]:LOW FLO. The input connector ground is not completely isolated from the chassis ground.

If the channel specifier is not used, the command defaults to channel 1.

**INPut[1 | 2 | 3 | 4]:REFerence:DIRection****command/query**

Sets the direction for the transducer point.

Command Syntax:           INPut [1 | 2 | 3 | 4]:REFerence:DIRection <number> | <bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"inp:ref:dir 3"  
                          OUTPUT 711;"INPUT:REFERENCE:DIRECTION 1"

Query Syntax:            INPut [1 | 2 | 3 | 4]:REFerence:DIRection?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

Description:

This command allows you to document the directional placement of the transducer on the device under test. Directions available are:

- 0 = no direction
- 1 = X
- 2 = Y
- 3 = Z
- 4 = (radial)
- 5 = T (tangential  $\theta \angle$ )
- 6 = P (tangential  $\theta \phi$ )
- 7 = TX
- 8 = TY
- 9 = TZ

The query response provides the current direction associated with the specified channel.

**INPut[1 | 2 | 3 | 4]:REFerence:POINt****command/query**

Sets the number for the transducer point.

Command Syntax:           INPut [1 | 2 | 3 | 4] :REFerence:POINt <number> | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: 0:32767

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":input:ref:poin 11081"  
                          OUTPUT 711;":INP:REF:POIN 12720"

Query Syntax:            INPut [1 | 2 | 3 | 4] :REFerence:POINt?

Return Format:           Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

Description:

This command allows you to specify a number for the point at which the transducer is attached to the device under test.

The query returns the current point number associated with the input channel.

**INPut[1 | 2 | 3 | 4][:STATe]****command/query**

Specifies one-channel, two-channel or four-channel measurements.

Command Syntax:           INPut [1 | 2 | 3 | 4] [:STATe] OFF | 0 | ON | 1

Example Statements:       OUTPUT 711;":inp2 OFF"  
                          OUTPUT 711;"Input:Stat ON"

Query Syntax:             INPut [1 | 2 | 3 | 4] [:STATe] ?

Return Format:            Integer

Attribute Summary:       Option:   not applicable  
                          Synchronization Required:   no  
                          Preset State:   Standard                           Option AY6  
  INPut1 ON (+1)                   INPut1 ON (+1)  
  INPut2 ON (+1)                   INPut2 ON (+1)  
  INPut3 not available       INPut3 OFF (+0)  
  INPut4 not available       INPut4 OFF (+0)  
                          SCPI Compliance:   confirmed

Description:

To select a two-channel measurement, send INP2 ON. With Option AY6, send INP2 ON;:INP4 OFF.

To select a one-channel measurement, send INP2 OFF. The analyzer takes data from Channel 1 only.

To select a four-channel measurement (only available with Option AY6), send INP4 ON.

---

**Note**                    INP1 OFF is not a valid command. The Channel 1 input can not be disabled.

---



**INSTRument**

---





## INSTrument

---

The commands in this subsystem select the instrument mode of the analyzer. Instrument mode specifies the type of measurement being made and whether signals applied to the front panel input connections are being measured.

Instrument mode is a major selection that changes the “personality” of the analyzer. This means that other parameters change when you change instrument mode.

---

**Note**            The analyzer “remembers” a separate set of parameters for each instrument mode.

---

**INSTrument:NSElect****command/query**

Selects one of the analyzer's six major instrument modes.

Command Syntax:           INSTrument:NSElect <number>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:5

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"INST:NSELECT 4"  
                          OUTPUT 711;"inst:nselect 4"

Query Syntax:             INSTrument:NSElect?

Return Format:            Integer

Attribute Summary:       Option: 1D0 Computed Order Tracking  
                          1D1 Realtime Octave  
                          1D2 Swept Sine  
                          Synchronization Required: no  
                          Preset State: 0 (FFT)  
                          SCPI Compliance: confirmed

Description:

The following commands select a major instrument mode:

- INST:NSEL 0 – FFT Analysis
- INST:NSEL 4 – Histogram
- INST:NSEL 5 – Correlation Analysis

The following commands are valid if the appropriate option is installed:

- INST:NSEL 1 – Octave Analysis – Option 1D1
- INST:NSEL 2 – Order Analysis – Option 1D0
- INST:NSEL 3 – Swept Sine – Option 1D2

The GPIB command set changes with each instrument mode. Parameters sent to setup a measurement for each instrument mode do not affect setups for other instrument modes. As a result, you should select the instrument mode near the beginning of any program sequence that defines the instrument state.

The default instrument mode is FFT at powerup and reset.

**INSTrument[:SElect]****command/query**

Selects one of the analyzer's six major instrument modes.

Command Syntax:           INSTrument [:SElect] FFT|OCTave|ORDer|SINE|HISTo-  
gram|CORRelation

Example Statements:        OUTPUT 711;":Inst SINE"  
OUTPUT 711;"INST:SELECT FFT"

Query Syntax:             INSTrument [:SElect] ?

Return Format:             FFT|OCT|ORD|SINE|HIST|CORR

Attribute Summary:        Option:   1D0 Computed Order Tracking  
                              1D1 Realtime Octave  
                              1D2 Swept Sine  
Synchronization Required: no  
Preset State:            FFT  
SCPI Compliance:        confirmed

Description:

The following commands select a major instrument mode:

- INST:SEL CORR – Correlation Analysis
- INST:SEL FFT – FFT Analysis
- INST:SEL HIST – Histogram

The following commands are valid if the appropriate option is installed:

- INST:SEL OCT – Octave Analysis – Option 1D1
- INST:SEL ORD – Order Analysis – Option 1D0
- INST:SEL SINE – Swept Sine – Option 1D2

The GPIB command set changes with each instrument mode. Parameters sent to setup a measurement for each instrument mode do not affect setups for other instrument modes. As a result, you should select the instrument mode near the beginning of any program sequence that defines the instrument state.



MEMory

---



## MEMory

---

This subsystem contains commands which manage instrument memory. This excludes memory used for mass storage, which is defined in the MMEMory subsystem.

**MEMory:CATalog[:ALL]?****query**

Returns information on the current contents and state of the analyzer's memory.

Query Syntax: MEMory:CATalog[:ALL]?

Example Statements: OUTPUT 711;"mem:catalog:all?"  
OUTPUT 711;"Mem:Catalog?"

Return Format: <Bytes\_in\_use>,<bytes\_available>,<ITEM>,<ITEM>,  
<ITEM>,<ITEM>,<ITEM>

<bytes\_in\_use> ::= total amount of memory currently allocated, in bytes  
<bytes\_available> ::= largest memory block currently available, in bytes  
<ITEM> ::= <NAME>,<TYPE>,<SIZE>  
<NAME> ::= TCAPture  
          WATERfall  
          WREG (waterfall register)  
          PROG (Instrument BASIC program)  
          RDISK (volatile RAM disk)  
          LIM (Limit lines)  
<TYPE> ::= BIN (binary file)  
<SIZE> ::= size of file in bytes

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: confirmed

Description:

Use this query to determine the analyzer's memory usage.

The analyzer allows you to allocate memory for the following items:

- time capture buffer
- waterfall display
- waterfall registers
- Instrument BASIC programs
- the analyzer's volatile RAM disk
- limit lines

This query returns a directory list and memory sizes for these items. The query returns <NAME> in short form (for example, TCAP for time capture).



**MEMory:CATalog:NAME?****query**

Returns information about memory usage allocated for a specific item.

Query Syntax:           MEMory:CATalog:NAME? TCAPture|WATerfall|WREGister|PRO-  
Gram|RDISK|LIMits

Example Statements:     OUTPUT 711;"MEMORY:CAT:NAME? WREGISTER"  
OUTPUT 711;"mem:cat:name? wat"

Return Format:           <NAME>

```
<NAME> ::= TCAP (time capture buffer)
          WAT (waterfall)
          WREG (waterfall register)
          PROG (Instrument BASIC program)
          RDIS (volatile RAM disk)
```

Attribute Summary:     Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

## Description:

This query returns the amount of the memory allocated for a specific item.

To allocate memory for a time capture buffer, send the [SENSe:]TCAPture:MALLocate command.

To allocate memory for a waterfall display, send the CALCulate:WATerfall:COUNT command.

The analyzer allocates memory for a waterfall register when you send the TRACe:WATerfall[:DATA] command.

The analyzer allocates memory for an Instrument BASIC program when you create the program. The memory usage is allocated for all Instrument BASIC programs. If multiple programs reside within the analyzer, you can not determine the memory allocated for a single Instrument BASIC program with this command.

To allocate memory for the volatile RAM disk, send the MMEMory:INITialize command.

The query returns <NAME> in short form (for example, TCAP for time capture).

---

**MEMory:DElete:ALL****command**

Purges all allocated memory in the analyzer.

Command Syntax:           MEMory:DElete:ALL

Example Statements:        OUTPUT 711;":MEM:DELETE:ALL"  
                              OUTPUT 711;"mem:delete:all"

Attribute Summary:         Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  confirmed

Description:

This command purges the analyzer's allocated memory for all the following items:

- time capture buffer
- waterfall display
- waterfall registers
- Instrument BASIC programs
- the analyzer's volatile RAM disk
- limit lines

The memory is available for reuse.

---

**MEMory:DELeTe[:NAME]****command**

Purges the memory allocated for a specific item.

Command Syntax:           MEMory:DELeTe[:NAME] TCAPture|WATerfall|WREGister|PRO-  
Gram|RDISK

Example Statements:        OUTPUT 711;"MEM:DEL WAT"  
                          OUTPUT 711;"mem:delete:name rdisk"

Attribute Summary:        Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

This command deletes the item. The analyzer's memory is available for reuse.

**MEMory:FREE[:ALL]?****query**

Returns information on the state of the analyzer's memory.

Query Syntax:               MEMory:FREE[:ALL]?

Example Statements:        OUTPUT 711;"Memory:Free:All?"  
                              OUTPUT 711;"MEMORY:FREE?"

Return Format:              <NR1>, <NR1>

                  <NR1>, <NR1> ::= <bytes\_available>, <bytes\_in\_use>  
<bytes\_available> ::= largest memory block currently available, in bytes  
<bytes\_in\_use>    ::= total amount of memory currently allocated, in bytes

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This query returns two values.

The first value specifies the amount of memory currently available for allocation. The second value specifies the total amount of memory currently allocated for the following items.

- time capture buffer
- waterfall display
- waterfall registers
- Instrument BASIC programs
- the analyzer's volatile RAM disk

MMEMory

---



## MMEMory

---

Commands in this subsystem control the analyzer's mass storage (disk) functions. Two of the mass storage devices are RAM-based disks—one using non-volatile RAM and the other using volatile RAM. Another mass storage device is an internal disk drive that uses 3.5 inch flexible disks. In addition, the analyzer can access an external disk drive. The disk drives (internal or external) can access either LIF or DOS disk formats. A maximum of five units can be active at any one time.

In most cases, if you do not send a mass storage specifier with a command that requires one, a default specifier is assumed. You select the default specifier with the MMEM:MSIS command.

### Syntax Descriptions

Syntax descriptions in this chapter use the following conventions:

<FILE> ::=

This string is used to describe the name of a file. It does not include any disk drive information.

The valid character set for <FILE> depends on the disk format.

DOS file names are limited to 8 ASCII characters followed by a period and a three character extension. The period and extension are not required. File names are not case sensitive.

LIF file names are limited to 10 character which may include any character except “:” “,” and “|”. The first character must be a letter. File names are case sensitive.

<FILENAME> ::=

This string is used to describe the name of a file on the default disk or on the specified disk. The allowed form is:

‘[PATH]filename’

where PATH must be replaced with:

“[MSIS:][\DOS\_DIR\]”

where MSIS: must be replaced with:

RAM: which selects volatile RAM.

NVRAM: which selects non-volatile RAM.

INT: which selects the internal disk drive.

EXT[,<select\_code>[,<unit\_number>]]: which selects the external disk drive.

The brackets are not literal in the parameter. They indicate that the disk drive designation is optional. If the disk drive is not specified, the default disk drive is used. Select the default specifier with the MMEM:MSIS command.

---

**Note** The <select\_code> and <unit\_number> specified with the MMEM:MSIS command becomes the default disk and unit address for the external disk drive (EXT:).

---

DOS\_DIR specifies a directory name (DOS only). DOS directory names are limited to ASCII characters. The forward slash (/) may be used as a directory separator instead of the backward slash (\). The brackets are not literal in the parameter. They indicate that the directory designation is optional. If the director is not specified, the default directory is used. Select the default specifier with the MMEM:MSIS command.

The valid character set for <FILENAME> depends on the disk format.

DOS file names are limited to 8 ASCII characters followed by a period and a three character extension. The period and extension are not required. File names are *not case sensitive*.

LIF file names are limited to 10 character which may include any character except “:” “,” and “|”. The first character must be a letter. File names are *case sensitive*.

<MSINAME> ::=

This parameter specifies the mass storage device. It takes one of the following values:

‘RAM:’  
‘NVRAM:’  
‘INT:’  
‘EXT[,<select\_code>[,<unit\_number>]]:’

The brackets are not literal in the parameter. They indicate that the external disk drive’s address is optional. If the <select\_code> and <unit\_number> are not specified, the analyzer uses the default values specified with the MMEM:MSIS command.

<MMEMNAME> ::=

This parameter specifies a single file or an entire mass storage device. It takes the same form as <FILENAME> except either the name of the file, the name of the mass storage device or both must be present. It can take one of the following forms:

‘[MSIS:][DOS\_DIR:]filename’  
‘MSIS:’  
‘MSIS:[DOS\_DIR:]filename’

See the description for <FILENAME> for the valid mass storage device specifiers and the valid file name character set.



**MMEMory:COPY****command**

Copies the contents of one disk to another or one file to another.

Command Syntax:           MMEMory:COPY <MMEMNAME>, <MMEMNAME>

```

<MMEMNAME> ::= '<disk>[<dos_dir>][<filename>]'
  <disk> ::= NVRAM:|RAM:|INT:|EXT[,<select_code>[,<unit_number>]]:
<select_code> ::= a real number (NRf data)
                  limits: 700:730
<unit_number> ::= a real number (NRf data)
                  limits: 0:3
  <dos_dir> ::= directory_name
              (DOS only)
<directory_name> ::= ASCII characters
<filename> ::= ASCII characters (see beginning of chapter
                  for <filename> restrictions)

```

Example Statements:       OUTPUT 711; "mmem:copy 'int:file1', 'ext:file1'"  
                           OUTPUT 711; "MMEM:COPY 'RAM:', 'INT:'"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: yes  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

The first <MMEMNAME> is the source; the second is the destination.

To copy a disk, use the disk specifier for each <MMEMNAME>. The select code and unit number specifiers are valid only with the EXT: disk specifier.

---

**Caution**           All files on the destination disk are overwritten when you specify a disk copy.

---

To copy a file, use disk specifiers and filenames. If you want to rename a file, use the MMEM:MOVE command. You can specify a filename as the source and a disk for the destination. The destination file has the same name as the source.

---

**Note**               When accessing the external mass storage device (EXT:); the active controller on the GPIB must temporarily pass control to the analyzer. After the command has been executed, the analyzer must pass control back. See the GPIB Programmer's Guide for more information on passing control.

---

**MMEMory:DElete****command**

Deletes one file or the contents of an entire disk.

Command Syntax:           MMEMory:DElete <MMEMNAME>

```

<MMEMNAME> ::= '<disk>[<dos_dir>][<filename>]'
<disk> ::= NVRAM:|RAM:|INT:|EXT[,<select_code>[,<unit_number>]]:
<select_code> ::= a real number (NRf data)
                 limits: 700:730
<unit_number> ::= a real number (NRf data)
                 limits: 0:3
<dos_dir> ::= directory_name
            (DOS only)
<directory_name> ::= ASCII characters
<filename> ::= ASCII characters (see beginning of chapter
                 for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"MMEMORY:DEL 'INT:JUNK' "  
                           OUTPUT 711;"mmem:del 'ext:state.sta' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: yes, for EXT only  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

To delete all files from a mass storage device only specify the <disk>. The select code and unit number specifiers are valid only with the EXT: disk specifier.

You can use the "\*" as a wildcard (for example, '\*.DAT').

---

**Note**           When accessing the external mass storage device (EXT:); the active controller on the GPIB must temporarily pass control to the analyzer. After the command has been executed, the analyzer must pass control back. See in the GPIB Programmer's Guide for more information on passing control.

---

**MMEMory:DISK:ADDRess****command/query**

Tells the analyzer which GPIB address is assigned to your external disk.

Command Syntax:           MMEMory:DISK:ADDRess <number>|<step>|<bound>

<number> ::= a real number (Nrf data)  
          limits: 0:30  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;"MMEMORY:DISK:ADDRESS 21"  
                          OUTPUT 711;"mmem:disk:addr 6"

Query Syntax:             MMEMory:DISK:ADDRess?

Return Format:            Integer

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

**Description:**

When you initiate an external disk operation with one of the MMEMory commands, the analyzer expects to find an external disk at the GPIB address specified with MMEM:DISK:ADDR. If an external disk is not at the specified address, the operation is automatically aborted.

**MMEMory:DISK:UNIT****command/query**

Specifies the unit of the external disk drive.

Command Syntax:           MMEMory:DISK:UNIT <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:10

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"MMEMORY:DISK:UNIT 0"  
                          OUTPUT 711;"mmem:disk:unit 1"

Query Syntax:             MMEMory:DISK:UNIT?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

---

**MMEMory:FSYSstem?****query**

Returns the type of file system for the default disk.

Query Syntax:                   MMEMory:FSYSstem?

Example Statements:            OUTPUT 711;":mmemory:fsys?"  
                                  OUTPUT 711;"Mmem:Fsystem?"

Return Format:                 LIF|DOS

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: instrument-specific

Description:

This query returns the type of file system on the default disk. To specify the default disk, use the MMEMory:MSIS command.

**MMEMory:INITialize****command**

Formats the specified disk.

Command Syntax:           MMEMory:INITialize [<MSINAME>], [{LIF|DOS}],

[<format\_option>], [<interleave\_factor>]

```

    <MSINAME> ::= '<disk>'
    <disk> ::= NVRAM:|RAM:|INT:|EXT[,<select_code>[,<unit_number>]]:
    <select_code> ::= a real number (NRf data)
                    limits: 700:730
    <unit_number> ::= a real number (NRf data)
                    limits: 0:3
    <format_option> ::= <number>
    <number> ::= a real number (NRf data)
                limits: 0:7000064 (see description below)
    <interleave_factor> ::= <number>
    <number> ::= a real number (NRf data)
                limits: 0:256

```

Example Statements:       OUTPUT 711; "MMEM:INIT 'RAM:', DOS, 100000"  
                           OUTPUT 711; "mmemory:initialize 'int:', lif, 0, 3"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: yes  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

The select code and unit number specifiers are valid only with the EXT: disk specifier.

The first <number> specifies the format option. You can specify the format option for the internal disk (INT:) and an external disk (EXT:). In NVRAM, the format option is ignored. In RAM the memory size is already specified in bytes. Use the <format\_option> field to specify memory size in bytes.

The <number> you enter after a floppy disk specifier is actually an encoded value that determines the disk's formatted capacity in kilobytes. See table 15-1.

The second <number> is the interleave factor. The default value is 0.

---

**Note**                   When accessing the external mass storage device (EXT:); the active controller on the GPIB must temporarily pass control to the analyzer. After the command has been executed, the analyzer must pass control back. See the GPIB Programmer's Guide for more information on passing control.

---

Table 15-1. Flexible Disk Format Options

Media	Format Option	Bytes/Sector	Sectors/Track	Tracks/Surface	Maximum Capacity (bytes)
1-MByte	0	256	16	77	630,784
	1*	256	16	77	630,784
	2	512	9	77	709,632
	3	1,024	5	77	788,480
	4**	256	16	77	270,336
	16	512	9	80	737,280
2-MByte	0	256	32	77	1,261,568
	1***	256	32	77	1,261,568
	2	512	18	77	1,419,264
	3	1,024	10	77	1,576,960
	4***	256	32	77	1,261,568
	16	512	18	80	1,474,560

\* Same as Option 0 (default) when using 1-MByte media.

\*\* Not supported in internal disk drive (INT:).

\*\*\* Same as Option 0 (default) when using 2-MByte

**MMEMory:LOAD:CFIT****command**

Loads a curve fit table into the analyzer from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:CFIT <FILENAME>

```

<FILENAME> ::= `[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{<DOS_DIR>}]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"MMEMORY:LOAD:CFIT 'CFFILE.FIT' "  
                           OUTPUT 711;"mmem:load:cfit 'int:synth1.syn' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command loads a curve fit table into the curve fit buffer. The file must have been saved with the MMEM:STOR:CFIT or MMEM:STOR:SYNT commands. We recommend saving curve fit files with the “.FIT” file extension and synthesis files with the “.SYN” file extension.

If you are loading a synthesis table into the curve fit buffer, the table must be in pole-zero format. See the CALC:SYNT:TTYP command for information about table formats.

The current curve fit table is overwritten.

---

**Note**                   The file extensions “.FIT” and “.SYN” are naming conventions. The file type is created with the MMEM:STOR:CFIT or MMEM:STOR:SYNT commands. This file-type data is embedded within the file.

---



**MMEMory:LOAD:CONTinue****command/query**

Continues the load operation of time capture and waterfall files saved on multiple disks.

Command Syntax:	MMEMory:LOAD:CONTinue
Example Statements:	OUTPUT 711;"MMEM:LOAD:CONT" OUTPUT 711;"mmem:load:cont"
Query Syntax:	MMEMory:LOAD:CONTinue?
Return Format:	Integer
Attribute Summary:	Option: not applicable Synchronization Required: no Preset State: not applicable SCPI Compliance: instrument-specific

**Description:**

This command loads split files which were saved with the MMEM:STOR:CONT commands. MMEM:LOAD:CONT is valid only with time capture and waterfall files.

Use the MMEM:LOAD:TCAP or MMEM:LOAD:WAT command to begin the load operation.

When the analyzer has completed loading the first split file, filename\_1, it generates a message, "Media full; Insert next disk with 'filename\_2'. Insert the disk containing filename\_2. Send this command to continuing loading the file.

You can use the MMEM:LOAD:CONT? to verify that the time capture or waterfall file has been transferred. If the MMEM:LOAD:CONT? query returns a +1, the analyzer has not completed the transfer. The query returns a 0 when the analyzer has transferred the entire time capture or waterfall file.

For additional information on the generation of split files, see the MMEM:STOR:CONT command.

**MMEMory:LOAD:DTABLE:TRACe{1 | 2 | 3 | 4}****command**

Loads a data table into the analyzer from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:DTABLE:TRACe1|2|3|4}, <FILENAME>

```

<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#123;DOS_DIR&#125;]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \&#123;DIR_NAME&#125;\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"mmem:load:dtab:trac2, 'int:testdtab' "  
                           OUTPUT 711;"MMEM:LOAD:DTAB:TRAC1, 'MYTABLE' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

The trace specifier selects which trace you are loading the data table into—TRAC1 for trace box A, TRAC2 for trace box B, TRAC3 for trace box C, and TRAC4 for trace box D. The parameter specifies the disk and filename. If the disk is not specified, the file is loaded from the default disk.

The file must have been saved with the MMEM:STOR:DTABLE command.

**MMEMory:LOAD:LIMit:LOWer:TRACe[1 | 2 | 3 | 4]****command**

Loads a lower limit for the specified trace from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:LIMit:LOWer:TRACe [1 | 2 | 3 | 4] <FILENAME>

```
<FILENAME> ::= `[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,&#106;<select_code>[,&#106;<unit _number>]]:
<DOS_DIR> ::= \&#106;DIR_NAME\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#106;filename> restrictions)
```

Example Statements:       OUTPUT 711;"Mmemory:Load:Limit:Low:Trace2 'limit1'"  
                           OUTPUT 711;"MMEM:LOAD:LIM:LOW:TRACE 'INT:LIM2'"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command loads the contents of a file into the lower limit register of the specified trace. The file must have been saved either with the MMEM:STOR:LIM:UPP, MMEM:LIM:LOW, or MMEM:STOR:TRAC command.

The trace specifier selects the trace—TRAC1 for trace A, TRAC2 for trace B, TRAC3 for trace C, or TRAC4 for trace D. Trace A is the default if you do not specify a trace. The parameter specifies the source.

Additional limit commands are available under CALCulate:LIMit and DISPlay[:WINDow[1|2|3|4]]:LIMit.

**MMEMory:LOAD:LIMit:UPPer:TRACe[1 | 2 | 3 | 4]****command**

Loads an upper limit for the specified trace from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:LIMit:UPPer:TRACe [1 | 2 | 3 | 4] <FILENAME>

```
<FILENAME> ::= `[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#123;DOS_DIR&#125;]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#123;<select_code>[,&#123;<unit _number>]&#125;]:
<DOS_DIR> ::= \&#123;DIR_NAME&#125;\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter for
              &#123;filename&#125; restrictions)
```

Example Statements:        OUTPUT 711;"Mmemory:Load:Limit:Upp:Trace2 'TRACE1' "  
                          OUTPUT 711;"MMEM:LOAD:LIM:UPP:TRACE 'INT:MYTRACE' "

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

This command loads the contents of a file into the upper limit register of the specified trace. The file must have been saved either with the MMEM:STOR:LIM:UPP, MMEM:STOR:LIM:LOW, or MMEM:STOR:TRAC command.

The trace specifier selects the trace—TRAC1 for trace A, TRAC2 for trace B, TRAC3 for trace C, or TRAC4 for trace D. Trace A is the default if you do not specify a trace. The parameter specifies the source.

Additional limit commands are available under CALCulate:LIMit and DISPlay[:WINDow[1|2|3|4]]:LIMit.

**MMEMory:LOAD:MATH****command**

Loads a complete set of math definitions into the analyzer from the specified disk.

Command Syntax:           MMEMory:LOAD:MATH <FILENAME>

```

<FILENAME> ::= '['<PATH>]<filename>'
<PATH> ::= <MSIS>[&#123;&#123;DOS_DIR&#125;&#125;]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \&#123;&#123;DIR_NAME&#125;&#125;\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"MMEM:LOAD:MATH 'EXT:MATHF1' "  
                           OUTPUT 711;"mmemory:load:math 'mymath' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command loads the contents of a file into the analyzer's function registers (F1 through F5) with math functions and the analyzer's constant registers (K1 through K5) with values. The file must have been saved with the MMEM:STOR:MATH command.

**MMEMory:LOAD:PROGram****command**

Loads an Instrument BASIC program into the analyzer from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:PROGram <FILENAME>

```

<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#106;<select_code>[,&#106;<unit _number>]]:
<DOS_DIR> ::= \&#106;DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#106;filename> restrictions)

```

Example Statements:       OUTPUT 711;"memory:load:program 'myprog' "  
                           OUTPUT 711;"MMEM:LOAD:PROG 'INT:IBFILE' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

## Description:

This command loads an Instrument BASIC program into the selected program buffer.

To specify the active program buffer, send the PROG:NAME command before sending this command. If a program buffer is not specified, the Instrument BASIC program loads into Program 1.

To load an Instrument BASIC program directly from your controller, use the PROGram[:SELEcted]:DEFine command.

**MMEMory:LOAD:STATe****command**

Loads an instrument state into the analyzer from the specified disk.

Command Syntax:           MMEMory:LOAD:STATE <number>|<bound>, <FILENAME>

```

<number> ::= a real number (NRf data)
           limits: 1:1
<bound>  ::= MAX|MIN
<FILENAME> ::= '['<PATH>]<filename>'
<PATH>   ::= <MSIS>[<DOS_DIR>]
<MSIS>   ::= RAM:|NVRAM:|INT:|
           EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
             (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
             for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"MMEM:LOAD:STAT 1, 'INT:STATE.STA'"  
                           OUTPUT 711;"mmemory:load:state 1, 'ext:mystate'"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

This command uses the contents of a file to redefine the instrument state. The file must have been saved with the MMEM:STOR:STAT command.

---

**MMEMory:LOAD:SYNThesis****command**

Loads a synthesis table into the analyzer from a file on the specified disk.

Command Syntax:           MMEMory:LOAD:SYNThesis <FILENAME>

```
<FILENAME> ::= '['<PATH>']<filename>'
<PATH> ::= <MSIS>[{'DOS_DIR'}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)
```

Example Statements:       OUTPUT 711;"memory:load:synthesis 'syfile.syn' "  
                          OUTPUT 711;"MMEM:LOAD:SYNT 'INT:FILE1.fit' "

Attribute Summary:       Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

This command loads a synthesis table into the curve fit buffer. The file must have been saved with the MMEM:STOR:CFIT or MMEM:STOR:SYNT commands.

The current synthesis table is overwritten.

---

**Note**           The file extensions “.FIT” and “.SYN” are naming conventions. The file type is created with the MMEM:STOR:CFIT or MMEM:STOR:SYNT commands. This file type data is embedded within the file.

---



**MMEMory:LOAD:TCAPture****command**

Loads a time capture file from the specified disk.

Command Syntax:           MMEMory:LOAD:TCAPture <FILENAME>

```

<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#123;&#123;DOS_DIR&#125;&#125;]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#123;&#123;<select_code>[,&#123;&#123;<unit_number>]]:
<DOS_DIR> ::= \&#123;&#123;DIR_NAME&#125;&#125;\
<DIR_NAME> ::= ASCII characters
               (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
               for <filename> restrictions)

```

Example Statements:        OUTPUT 711;"MMEM:LOAD:TCAPTURE, 'MYTIME' "  
                           OUTPUT 711;"mmem:load:tcap, 'int:tcap1' "

Attribute Summary:        Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command loads the contents of a file into the time capture buffer. The file must have been saved with the MMEM:STOR:TCAP command. To display the time capture buffer use the CALC:FEED 'TCAP[1|2|3|4]' command.

Use the MMEM:LOAD:CONT if the file is saved on multiple disks. See the MMEM:LOAD:CONT command for more information about loading split files on multiple disks.

**MMEMory:LOAD:TRACe****command**

Loads a trace into the analyzer from the specified disk.

Command Syntax:           MMEMory:LOAD:TRACe {D1|D2|D3|D4|D5|D6|D7|D8}, <FILE-NAME> [, <NOSCALE>]

```
<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)
```

Example Statements:       OUTPUT 711;"MMEM:LOAD:TRAC D4, 'MYTRACE' "  
                          OUTPUT 711;"mmem:load:trace D5, 'int:testtr' "

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

## Description:

This command loads the contents of a file into one of the analyzer's eight data registers (D1 through D8). The first parameter specifies the destination. The second parameter specifies the source.

The optional parameter, <NOSCALE>, specifies the scaling of the trace. If NOSCALE is not included, the data register is scaled to match the scaling of the trace when it was saved to the file. If the NOSCALE parameter is included, the current scaling of the data register is not modified. This parameter is not compatible with SCPI.

The file must have been saved with the MMEM:STOR:TRAC command. After loading the data register you can display its contents with the CALC:FEED 'D{1|2 . . . |8}' command.

**MMEMory:LOAD:WATerfall****command**

Loads a waterfall file into the analyzer from a file on the specified disk.

Command Syntax:           MEMory:LOAD:WATerfall {W1|W2|W3|W4|W5|W6|W7|W8}, <FILE-NAME>

```
<FILENAME> ::= '['<PATH>]<filename>'
<PATH> ::= <MSIS>[&{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)
```

Example Statements:       OUTPUT 711;"MMEMORY:LOAD:WAT W8, 'INT:MYWAT' "  
                          OUTPUT 711;"mmem:load:wat w3, 'testwat' "

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

This command loads the contents of a file into one of the analyzer's eight waterfall registers (W1 through W8). The file must have been saved with the MMEM:STOR:WAT command.

After loading the data register you can display its contents with the CALC:FEED 'W{1|2 . . . |8}' command.

Use the MMEM:LOAD:CONT if the file is saved on multiple disks. See the MMEM:LOAD:CONT command for more information about loading split files on multiple disks.

**MMEMory:MDIRectory****command**

Creates a directory on a DOS file system.

Command Syntax:           MMEMory:MDIRectory <DIR\_NAME>

```
<DIR_NAME> ::= '['<disk>']\<directory_name>'
<disk> ::= NVRAM:|RAM:|INT:|EXT[,<select_code>[,<unit_number>]]:
           (MS-DOS file systems only)
<select_code> ::= a real number (NRf data)
                 limits: 700:730
<unit_number> ::= a real number (NRf data)
                 limits: 0:3
<directory_name> ::= ASCII characters
```

Example Statements:       OUTPUT 711;"Mmemory:Mdir `ram:\examp\test1`"  
                          OUTPUT 711;"MMEM:MDIRECTORY `RESULTS`"

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**MMEMory:MOVE****command**

Renames a file.

Command Syntax:           MMEMory:MOVE &lt;OLDNAME&gt;, &lt;NEWNAME&gt;

&lt;OLDNAME&gt; ::= '&lt;file&gt;'

<file> ::= ASCII characters (see beginning of chapter  
for <filename> restrictions)

&lt;NEWNAME&gt; ::= '&lt;file&gt;'

<file> ::= ASCII characters (see beginning of chapter  
for <file> restrictions)Example Statements:       OUTPUT 711;"mmemory:move `int:file1`, `myfile`"  
                          OUTPUT 711;"MMEM:MOVE `TESTFILE`, `FILE3`"Attribute Summary:       Option: not applicable  
                          Synchronization Required: yes  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

The &lt;OLDNAME&gt; is the old file name; &lt;NEWNAME&gt; is the new file name.

This command only allows you to change a file's name on the current disk. It does not allow you to transfer a file by changing the file's name and disk specifier. To transfer a file, first copy it to another disk with the MMEM:COPY command, then delete it from the original disk with the MMEM:DEL command.

---

**Note**                   When accessing the external mass storage device (EXT:); the active controller on the GPIB must temporarily pass control to the analyzer. After the command has been executed, the analyzer must pass control back. See the GPIB Programmer's Guide for more information on passing control.

---

**MMEMory:MSIS****command/query**

Specifies a default disk.

Command Syntax: MMEMory:MSIS &lt;MSINAME&gt;

```
<MSINAME> ::= '<disk>' | "<disk>"
<disk> ::= NVRAM: | RAM: | INT: | EXT[, <select_code>[, <unit_number>]]:
```

```
Example Statements: OUTPUT 711;":Mmem:Msis `INT:´"
OUTPUT 711;"MMEMORY:MSIS `RAM:´"
```

Query Syntax: MMEMory:MSIS?

Return Format: "NVRAM:" | "RAM:" | "INT:" | "EXT:"

```
Attribute Summary: Option: not applicable
Synchronization Required: no
Preset State: saved in non-volatile memory
SCPI Compliance: confirmed
```

Description:

If you omit disk specifiers from MMEMory commands, the commands are automatically directed to the default disk. This command uses the following mnemonics to select the default disk:

- NVRAM: — selects the non-volatile RAM disk.
- RAM: — selects the volatile RAM disk.
- INT: — selects the internal disk.
- EXT: — selects the external disk. The select code and unit number specifiers are valid only with the EXT: disk specifier. If the select code and unit number are specified with this command, they become the default for all external disks (EXT:) specifiers.

To determine the type of file system for the default disk, send the MMEM:FSYSstem? query.

---

**Note** When accessing the external mass storage device (EXT:); the active controller on the GPIB must temporarily pass control to the analyzer. After the command has been executed, the analyzer must pass control back. See the GPIB Programmer's Guide for more information on passing control.

---

**MMEMory:NAME****command/query**

Specifies a filename for the output of a print or plot operation.

Command Syntax:           MMEMory:NAME <FILENAME>

```
<FILENAME> ::= `[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#1060;DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,<select_code>[,<unit_number>]]:
<DOS_DIR> ::= \&#1060;DIR_NAME>
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see description for
              &#1060;filename> restrictions)
```

Example Statements:       OUTPUT 711;"mmemory:name `int:plot.hpg`"  
                           OUTPUT 711;"MMEM:NAME `PRINT1`"

Query Syntax:           MMEMory:NAME?

Return Format:           "STRING"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: approved

Description:

This command is used with the HCOP:DEST 'MMEM' command.

The valid character set for <filename> depends on the disk format.

DOS file names are limited to 8 ASCII characters followed by a period and a 3-ASCII-character extension. The period and extension are not required. File names are not case sensitive. An optional directory path is allowed. The forward slash (/) may be used as a directory separator instead of the backward slash (\). DIR\_NAME may be composed of a series of subdirectory names, separated by slash marks.

LIF file names are limited to 10 ASCII characters which may include all characters except ":", "<" and "|". The first character must be a letter. File names are case sensitive.

**MMEMory:STORe:CFIT****command**

Stores a curve fit table to a file on the specified disk.

Command Syntax:           MMEMory:STORe:CFIT <FILENAME>

                          <FILENAME> ::= '[<PATH>]<filename>'

                          <PATH> ::= <MSIS>[&#106;{<DOS\_DIR>}]

                          <MSIS> ::= RAM:|NVRAM:|INT:|

  EXT[,&#106;<select\_code>[,&#106;<unit \_number>]]:

                          <DOS\_DIR> ::= \&#106;<DIR\_NAME>\

                          <DIR\_NAME> ::= ASCII characters

  (DOS only)

                          <filename> ::= ASCII characters (see beginning of chapter

  for <filename> restrictions)

Example Statements:       OUTPUT 711;"MMEMORY:STORe:CFIT 'CFFILE.FIT'"

                          OUTPUT 711;"mmem:stor:cfit 'int:curve1.fit'"

Attribute Summary:       Option: not applicable

                          Synchronization Required: no

                          Preset State: not applicable

                          SCPI Compliance: instrument-specific

## Description:

This command stores the current curve fit table to a file. We recommend adding the file extension, ".FIT", to all curve fit files. See the DISP:CONT MMEM command for information about displaying the disk catalog.

If the file name matches the name of another file on the disk, this command overwrites the old file.



**MMEMory:STORe:CONTInue****command/query**

Splits a large file (a time capture file or a waterfall file) over multiple disks.

Command Syntax:	MMEMory:STORe:CONTInue
Example Statements:	OUTPUT 711;"mmem:store:cont" OUTPUT 711;"Mmem:Store:Cont"
Query Syntax:	MMEMory:STORe:CONTInue?
Return Format:	Integer
Attribute Summary:	Option: not applicable Synchronization Required: no Preset State: not applicable SCPI Compliance: instrument-specific

## Description:

This command splits files created by the MMEM:STOR:WAT and the MMEM:STOR:TCAP commands only.

If the buffer is too large for the disk when you send the MMEM:STOR:WAT or MMEM:STOR:TCAP command, an error, "Media full; File too large" is generated. Send the MMEM:STOR:CONT command to begin the save operation. (The error message is generated before the save operation is implemented).

MMEM:STOR:CONT adds a numeric specifier to the <filename>. Split files appear as filename\_1, filename\_2, etc. in the disk catalog (see VIEW:CONT MMEM).

MMEM:STORE CONT? returns a +1 if the analyzer has not completely saved the time capture file or waterfall file. Insert the new disk and send the MMEM:STORE:CONT command to continue the save operation. The query returns a 0 when the entire time capture or waterfall has been saved.

**MMEMory:STORe:DTABle:TRACe1|2|3|4****command**

Saves the specified data table to a file on the specified disk.

Command Syntax:           MMEMory:STORe:DTABle:TRACe1|2|3|4}, <FILENAME>

```
<FILENAME> ::= '['<PATH>]<filename>'
<PATH> ::= <MSIS>[&#123;&#123;DOS_DIR&#125;&#125;]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,&#123;&#123;select_code&#125;&#125;[,&#123;&#123;unit_number&#125;&#125;]]:
<DOS_DIR> ::= \&#123;&#123;DIR_NAME&#125;&#125;\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#123;&#123;filename&#125;&#125; restrictions)
```

Example Statements:       OUTPUT 711;"mmem:stor:dtab:trac2, 'int:testdtab'"  
                           OUTPUT 711;"MMEM:STORE:DTAB:TRAC1, 'MYTABLE' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

The trace specifier selects which data table you are saving—TRAC1 for the data table appearing in trace box A, TRAC2 for the data table appearing in trace box B, TRAC3 for the data table appearing in trace box C, TRAC4 for the data table appearing in trace box D. The parameter specifies the disk and filename. If the disk is not specified, the file is saved to the default disk.

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.

---

**Note**                   If you plan to transfer this file to a PC, refer to the Standard Data Format Utilities Users Guide .

---

**MMEMory:STORe:LIMit:LOWer:TRACe[1 | 2 | 3 | 4]****command**

Saves the lower limit of the specified trace to a file on the specified disk.

Command Syntax:           MMEMory:STORe:LIMit:LOWer:TRACe [1 | 2 | 3 | 4] <FILENAME>

```
<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,&#106;<select_code>[,&#106;<unit_number>]]:
<DOS_DIR> ::= \&#106;DIR_NAME\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#106;filename> restrictions)
```

Example Statements:       OUTPUT 711;Mmem:Store:Limit:Low:Trace 'newlim.lim' "  
                          OUTPUT 711;MMEM:STOR:LIM:LOW:TRAC2 'int:lim2' "

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

**Description:**

This command saves a lower limit to a file. The trace specifier selects which lower limit you are saving—TRAC1 for trace A, TRAC2 for trace B, TRAC3 for trace C, or TRAC4 for trace D. Trace A is the default if you do not specify a trace.

If the file name you specify matches the name of another file on the disk, this command overwrites the old file.

---

**Note**                   If you plan to transfer this file to a PC, refer to the Standard Data Format Utilities Users Guide .

---

**MMEMory:STORe:LIMit:UPPer:TRACe[1 | 2 | 3 | 4]****command**

Saves the upper limit of the specified trace to a file on the specified disk.

Command Syntax:           MMEMory:STORe:LIMit:UPPer:TRACe [1 | 2 | 3 | 4] <FILENAME>

```
<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[<DOS_DIR>]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)
```

Example Statements:        OUTPUT 711:mmem:store:limit:upper:trace2 'int:newlim' "  
                              OUTPUT 711;"MMEM:STOR::LIM:UPP:TRACE1, 'LIMIT1.LIM' "

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

**Description:**

This command saves an upper limit to a file. The trace specifier selects which upper limit you are saving—TRAC1 for trace A, TRAC2 for trace B, TRAC3 for trace C, or TRAC4 for trace D. Trace A is the default if you do not specify a trace.

If the file name you specify matches the name of another file on the disk, this command overwrites the old file.

---

**Note**                    If you plan to transfer this file to a PC, refer to the Standard Data Format Utilities Users Guide .

---

**MMEMory:STORe:MATH****command**

Saves a complete set of math definitions to a file on the specified disk.

Command Syntax:           MMEMory:STORe:MATH <FILENAME>

                  <FILENAME> ::= '['<PATH>]<filename>'

                  <PATH> ::= <MSIS>[&#106;{DOS\_DIR}]

                  <MSIS> ::= RAM:|NVRAM:|INT:|

                                  EXT[,&#106;<select\_code>[,&#106;<unit \_number>]]:

                  <DOS\_DIR> ::= \&#106;DIR\_NAME&#106;\

                  <DIR\_NAME> ::= ASCII characters

                                  (DOS only)

                  <filename> ::= ASCII characters (see beginning of chapter

                                  for &#106;filename&#106; restrictions)

Example Statements:       OUTPUT 711;"MMEM:STORe:MATH 'EXT:NEWMATH' "

                              OUTPUT 711;"mmemory:store:math 'mymath.def' "

Attribute Summary:       Option: not applicable

                              Synchronization Required: no

                              Preset State: not applicable

                              SCPI Compliance: instrument-specific

**Description:**

This command saves the math functions in the analyzer's five function registers (F1 through F5) and the current values in the analyzer's constant registers (K1 through K5) to a file.

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.

**MMEMory:STORe:PROGram****command**

Saves an Instrument BASIC program to a file on the specified disk.

Command Syntax:           MMEMory:STORe:PROGram <FILENAME>

```

<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{&#106;DOS_DIR}&#106;]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#106;select_code&#106;[,&#106;unit_number&#106;]]:
<DOS_DIR> ::= \&#106;DIR_NAME&#106;\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#106;filename&#106; restrictions)

```

Example Statements:        OUTPUT 711;"Mmemory:Stor:Prog 'IBPROG' "  
                           OUTPUT 711;"MMEM"STORE:PROG 'INT:MYPROG' "

Attribute Summary:        Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

## Description:

This command saves the currently active Instrument Basic program to the specified disk. The program must be located in the active program buffer (see the PROG:NAME command). If the active program buffer does not contain a program the analyzer generates an error, "Program Error, No program exists."

If the file name matches the name of another file on the disk, the this command overwrites the old file.

**MMEMory:STORe:PROGram:FORMat****command/query**

Specifies the format Instrument BASIC programs are stored.

Command Syntax:	MMEMory:STORe:PROGram:FORMat ASCii BINary
Example Statements:	OUTPUT 711;" :MMEM:STOR:PROG:FORM ASC" OUTPUT 711;"mmemory:store:program:format binary"
Query Syntax:	MMEMory:STORe:PROGram:FORMat?
Return Format:	ASC BIN
Attribute Summary:	Option: not applicable Synchronization Required: no Preset State: not applicable SCPI Compliance: instrument-specific

## Description:

To store an Instrument BASIC program in ASCII format, send MMEM:STOR:PROG:FORM ASC.

To store an Instrument BASIC program in internal binary representation, send MMEM:STOR:PROG:FORM BIN.

ASCII program take more time to load than binary programs. Binary programs take more storage space than ASCII programs.

**MMEMory:STORe:STATe****command**

Saves the instrument state to a file on the specified disk.

Command Syntax:           MMEMory:STORe:STATe <number>|<bound>, <FILENAME>

```

<number> ::= a real number (NRf data)
           limits: 1:1
<bound>  ::= MAX|MIN
<FILENAME> ::= '['<PATH>]<filename>'
<PATH>   ::= <MSIS>[<DOS_DIR>]
<MSIS>   ::= RAM:|NVRAM:|INT:|
           EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
             (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
             for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"memory:store:state1, 'ext:mystate'"  
                           OUTPUT 711;"MMEM:STOR:STAT 1, 'INT:STATE.STA'"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.



**MMEMory:STORe:SYNThesis****command**

Stores a synthesis table to a file on the specified disk.

Command Syntax:           MMEMory:STORe:SYNthesis <FILENAME>

```

<FILENAME> ::= '['<PATH>]<filename>'
<PATH> ::= <MSIS>[&{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,<select_code>[,<unit _number>]]:
<DOS_DIR> ::= \<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)

```

Example Statements:       OUTPUT 711;"mmem:store:synthesis 'int:syfile.syn' "  
                           OUTPUT 711;"MMEM:STOR:SYNT 'NEWSYNTH.SYN' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

## Description:

This command stores the current synthesis table to a file. We recommend adding the file extension, “.SYN”, to all synthesis files. See the DISP:CONT MMEM command for information about displaying the disk catalog.

If the file name matches the name of another file on the disk, this command overwrites the old file.

**MMEMory:STORE:TCAPture****command**

Saves the time capture buffer to a file on the specified disk.

Command Syntax:           MEMory:STORe:TCAPture <FILENAME>

```

<FILENAME> ::= `[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#106;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#106;<select_code>[,&#106;<unit _number>]]:
<DOS_DIR> ::= \&#106;DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#106;filename> restrictions)

```

Example Statements:       OUTPUT 711;"mmem:stor:tcap, 'int:tcap1'"  
                           OUTPUT 711;"MMEM:STORE:TCAPTURE, 'MYTIME' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command saves the current time capture buffer to a file on the specified disk.

If the buffer is too large for the disk, an error, "Media full; File too Large" is generated. Use the MMEM:STORE CONT command to split the file over multiple disks. See the MMEM:STORE CONT command for more information about splitting a time capture file on multiple disks.

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.

---

**Note**                    If you plan to transfer this file to a PC, refer to the MMEM:STOR:TRAC:FORM command.

---

**MMEMory:STORe:TRACe****command**

Saves the specified trace to a file on the specified disk.

Command Syntax:           MMEMory:STORe:TRACe {TRACe1|TRACe2|TRACe3|TRACe4}, <FILE-NAME>

```
<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#92;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
           EXT[,&#92;<select_code>[,&#92;<unit _number>]]:
<DOS_DIR> ::= \&#92;<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for &#92;<filename> restrictions)
```

Example Statements:       OUTPUT 711;"mmem:stor:trace D5, 'int:testtr'"  
                           OUTPUT 711;"MMEM:STORE:TRAC D4, 'MYTRACE' "

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

**Description:**

The first parameter specifies which trace you are saving—TRAC1 for the trace appearing in trace box A, TRAC2 for the trace appearing in trace box B, TRAC3 for the trace appearing in trace box C, and TRAC4 for the trace appearing in trace box D. The second parameter specifies the disk and filename. If the disk is not specified, the file is saved to the default disk.

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.

This command differs from the TRAC:DATA command as the MMEM:STOR:TRAC command only saves trace data to a file. TRAC:DATA saves trace data to one of the data registers.

---

**Note**                    If you plan to transfer this file to a PC, refer to the MMEM:STOR:TRAC:FORM command.

---

---

**MMEM:STORe:TRACe:FORMat****command/query**

Specify the format of data saved to a file.

Command Syntax:           MMEM:STORe:TRACe:FORMat   SDF|ASCII

Example Statements:       OUTPUT 711;"MMEM:STOR:TRAC:FORM ASCII"

Query Syntax:             MMEM:STORe:TRACe:FORMat?

Return Format:             SDF|ASCII

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset state: SDF  
                          SCPI Compliance: instrument-specific

Description:

Use this command to specify the format of traces, waterfalls, and time captures saved to a file.

Use the SDF (Standard Data Format) format if you want to save the data for later recall into the analyzer.

Use the ASCII format if you want to save the trace values as ASCII characters. This format is useful if you want to transfer the data to a PC.

**MMEMory:STORe:WATerfall****command**

Saves the current waterfall display to a file on the specified disk.

Command Syntax:           MMEMory:STORe:WATerfall {TRACe1 | TRACe2 | TRACe3 | TRACe4, <FILENAME>

```
<FILENAME> ::= '[<PATH>]<filename>'
<PATH> ::= <MSIS>[&#92;{DOS_DIR}]
<MSIS> ::= RAM:|NVRAM:|INT:|
          EXT[,&#92;<select_code>[,&#92;<unit _number>]]:
<DOS_DIR> ::= \&#92;<DIR_NAME>\
<DIR_NAME> ::= ASCII characters
              (DOS only)
<filename> ::= ASCII characters (see beginning of chapter
              for <filename> restrictions)
```

Example Statements:       OUTPUT 711;"MMEM:STORE:WATERFALL TRAC1, "LASTWAT'"  
                           OUTPUT 711;"mmem:stor:wat trace2, 'int:mywat'"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: instrument-specific

**Description:**

This command saves a waterfall display to a file. The waterfall buffer must contain more than one trace.

The first parameter specifies which trace you are saving—TRAC1 for the waterfall appearing in trace box A, TRAC2 for the waterfall appearing in trace box B, TRAC3 for the trace appearing in trace box C, or TRAC4 for the waterfall appearing in trace box D. The second parameter specifies the disk and filename. If the disk is not specified, the file is saved to the default disk.

If the filename you specify matches the name of another file on the disk, this command overwrites the old file.

If the waterfall display is too large for the disk, an error, "Media full; File too large." Use the MMEM:STORE CONT command to split the file over multiple disks. See the MMEM:STORE CONT command for more information about splitting a time capture file on multiple disks.

---

**Note**                    If you plan to transfer this file to a PC, refer to the MMEM:STOR:TRAC:FORM command.

---



OUTPut

---





## OUTPut

---

This subsystem contains two commands. One turns on the analyzer's source output and the other turns on a low pass filter for the optional arbitrary source (Option 1D4). See the SOURce subsystem for commands which define the analyzer's source output.

---

**OUTPut:FILTer[:LPASs][:STATe]**

**command/query**

Turns on a low pass filter for the arbitrary source.

Command Syntax:           OUTPut:FILTer[:LPASs][:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;"OUTP:FILT:LPASS:STAT OFF"  
                               OUTPUT 711;"output:filt ON"

Query Syntax:             OUTPut:FILTer[:LPASs][:STATe]?

Return Format:             OFF|ON

Attribute Summary:        Option: 1D4 Arbitrary Source  
                               Synchronization Required: no  
                               Preset State: OFF (+0)  
                               SCPI Compliance: confirmed

Description:

This command turns on a low pass filter for the arbitrary source that eliminates unwanted images. The analyzer sets the cutoff frequency beyond the stop frequency of the frequency span.

See online help for additional information.

**OUTPut[:STATe]****command/query**

Enables the analyzer's internal source.

Command Syntax:           OUTPut [:STATe] OFF|0|ON|1

Example Statements:        OUTPUT 711;":Outp ON"  
                          OUTPUT 711;"OUTPUT:STAT ON"

Query Syntax:             OUTPut [:STATe] ?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                          Synchronization Required:  no  
                          Preset State:  OFF (+0)  
                          SCPI Compliance:  confirmed

Description:

This command is not available for the swept sine analysis instrument mode (INST:SEL SINE).



PROGram

---



## PROGram

---

The commands in this subsystem are only available when the Instrument BASIC option is installed (Option 1C2). The commands in the PROGram subsystem allow you to generate and control Instrument BASIC programs in the analyzer.

The commands grouped under the SElected mnemonic operate on the active program buffer. Since SElected is an implied mnemonic, you can omit it from the PROGram commands. See “Implied Mnemonics” in chapter 2 of the GPIB Programmer’s Guide for more information.

The command under the EXPLicit mnemonic operates on any one of the analyzer’s five program buffers—not just the active program buffer.

**PROGram:EDIT:ENABle****command/query**

Disables the Instrument BASIC editor.

Command Syntax:           PROGram:EDIT:ENABle OFF|0|ON|1

Example Statements:        OUTPUT 711;"prog:edit:enab 1"  
                              OUTPUT 711;"PROG:EDIT:ENABLE OFF"

Query Syntax:             PROGram:EDIT:ENABle?

Return Format:             Integer

Attribute Summary:        Option: 1C2 Instrument BASIC  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: instrument-specific

**Description:**

This command disables and enables the Instrument BASIC edit menus. The edit menus are enabled when the analyzer is turned on.

To prevent access to the edit menus, send the PROG:EDIT:ENAB OFF command. The [EDIT] softkey is ghosted on the front panel display. An execution error is generated if the [EDIT] softkey is pressed while it is disabled. The message "The Instrument BASIC editor has been disabled." is displayed.

To enable the Instrument BASIC editor, send PROG:EDIT:ENAB ON.



**PROG:EXPL:DEF****command/query**

Loads an Instrument BASIC program into the specified program buffer from an external controller.

Command Syntax:           PROG:EXPL:DEF {PROG1 | PROG2 | PROG3 | PROG4 | PROG5}, <PROGRAM>

```

<PROGRAM> ::= <BLOCK>
<BLOCK> ::= #<byte>[<length_bytes>]<data_bytes>
<byte> ::= one byte specifying the number of length bytes
to follow (ASCII-encoded)
<length_bytes> ::= bytes specifying the number of data bytes
to follow (ASCII-encoded)
<data_bytes> ::= the bytes that define an Instrument BASIC program

```

## Example Statements:

```

Indefinite Block           OUTPUT 711;":PROG:EXPL:DEF PROG1,#0";
                          OUTPUT 711;"10 PRINT ""HELLO WORLD""&CHR$(10);
                          OUTPUT 711;"20 END"&CHR$(10);
                          OUTPUT 711;CHR$(10) END

```

```

Definite Block            OUTPUT 711;":PROG:EXPL:DEF PROG4,#230";
                          OUTPUT 711;"10 PRINT ""HELLO WORLD""&CHR$(10);
                          OUTPUT 711;"20 END"&CHR$(10);

```

Query Syntax:            PROG:EXPL:DEF?

Return Format:            definite length <BLOCK>

Attribute Summary:      Option: 1C2 Instrument BASIC  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: confirmed

## Description:

This command transfers a program between the analyzer and your controller. This allows you to develop a program on your controller and then load it into the analyzer. The first parameter specifies the program buffer. This becomes the active program buffer. The second parameter is the Instrument BASIC program.

When you transfer a program to the analyzer, you can use either the definite or the indefinite length block syntax. The simplest way to load an Instrument BASIC program into the analyzer is to send this command followed by #0, followed by all the characters making up the program (including line numbers and line feeds at the end of each program statement). Terminate the entire command with line feed character (ASCII decimal 10) and <^END> (the GPIB END message, EOI set true).

When the analyzer returns the program to your controller, it always uses the definite length block syntax. See "GPIB Message Syntax" in the GPIB Programmer's Guide for more information.

---

**PROGram:EXPLicit:LABel**

**command/query**

Loads a softkey label for the specified Instrument BASIC program.

Command Syntax:           PROGram:EXPLicit:LABel  
                               PROGram1 | PROGram2 | PROGram3 | PROGram4 | PROGram5, '<STRING>'

<STRING> ::= maximum of 18 ASCII characters  
                               (2 lines of 9 characters)

Example Statements:        OUTPUT 711;":Program:Expl:label Prog1,' START        TEST' "  
                               OUTPUT 711;"PROG:EXPL:LAB PROG5,' PRINT        REPORT' "

Query Syntax:             PROGram:EXPLicit:LABel?

Return Format:             "<STRING>"

Attribute Summary:        Option: 1C2 Instrument BASIC  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

Description:

This command allows you to customize the front panel softkey labels for Instrument BASIC programs.

**PROG[am]:SElecteD]:DEFine****command/query**

Loads an Instrument BASIC program from an external controller into the active program buffer.

Command Syntax:            PROG[am]:SElecteD]:DEFine <PROGRAM>

```
<PROGRAM> ::= <BLOCK>
<BLOCK> ::= #<byte>[<length_bytes>]<data_bytes>
<byte> ::= one byte specifying the number of length bytes
to follow (ASCII-encoded)
<data_bytes> ::= the bytes that define an Instrument BASIC program
```

**Example Statements:**

Indefinite Block            OUTPUT 711;":PROG:DEF #0";  
                               OUTPUT 711;"10 PRINT ""HELLO WORLD""&CHR\$(10);  
                               OUTPUT 711;"20 END"&CHR\$(10);  
                               OUTPUT 711;CHR\$(10) END

Definite Block             OUTPUT 711;":PROG:DEF #230";  
                               OUTPUT 711;"10 PRINT ""HELLO WORLD""&CHR\$(10);  
                               OUTPUT 711;"20 END"&CHR\$(10);

Query Syntax:             PROG[am]:SElecteD]:DEFine?

Return Format:             definite length <BLOCK>

Attribute Summary:       Option: 1C2 Instrument BASIC  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: confirmed

**Description:**

This command transfers a program between the analyzer and your controller. This allows you to develop a program on your controller and then load it into the analyzer.

Use the PROG[:SEL]:NAME to select the active program buffer.

When you transfer a program to the analyzer, you can use either the definite or the indefinite length block syntax. The simplest way to load an Instrument BASIC program into the analyzer is to send this command followed by #0, followed by all the characters making up the program (including line numbers and line feeds at the end of each program statement). Terminate the entire command with line feed character (ASCII decimal 10) and <^END> (the GPIB END message, EOI set true).

When the analyzer returns the program to your controller, it always uses the definite length block syntax. See "GPIB Message Syntax" in the GPIB Programmer's Guide for more information.

**PROGram[:SElected]:DElete:ALL**

**command**

Deletes all Instrument BASIC programs stored in the analyzer.

Command Syntax:            PROGram[:SElected]:DElete:ALL

Example Statements:        OUTPUT 711;":prog:delete:all"  
                              OUTPUT 711;"Prog:Selected:Del:All"

Attribute Summary:        Option: 1C2 Instrument BASIC  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

In addition to deleting the active program, this command deletes all of the resident Instrument BASIC programs. Program variables—both those in COM and those not in COM are deleted as well.

This is equivalent to a “Scratch A” operation.

---

**PROGrama[:SElected]:DElete[:SElected]****command**

Deletes the active Instrument BASIC program.

Command Syntax:           PROGrama[:SElected]:DElete[:SElected]

Example Statements:       OUTPUT 711;"PROG:SEL:DELETE:SEL"  
                          OUTPUT 711;"program:del"

Attribute Summary:       Option: 1C2 Instrument BASIC  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

In addition to deleting the active program, this command deletes all of the program variables—both those in COM and those not in COM. Specify the active program with the PROG:NAME command.

This is equivalent to a “Scratch A” operation.

**PROGram[:SElected]:LABel****command/query**

Loads a softkey label for the active Instrument BASIC program.

Command Syntax:           PROGram[:SElected]:LABel '<STRING>'

<STRING> ::= maximum of 18 ASCII characters  
(2 lines of 9 characters)

Example Statements:       OUTPUT 711;" :Program:label ` START       TEST' "  
                              OUTPUT 711;"PROG:SEL:LAB ` PRINT       REPORT' "

Query Syntax:             PROGram[:SElected]:LABel?

Return Format:            "<STRING>"

Attribute Summary:       Option: 1C2 Instrument BASIC  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

This command allows you to customize the front panel softkey labels for Instrument BASIC programs.

Specify the active program with the PROG:NAME command.

**PROGram[:SElected]:MAlLocate****command/query**

Allocates memory space for Instrument BASIC programs.

Command Syntax:           PROGram[:SElected]:MAlLocate {<number>|<bound>|DEFault}

<number> ::= a real number (NRf data)  
                  limits: 1200:500000

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":Prog:MAllocate 416211"  
                          OUTPUT 711;"PROG:SELECTED:MALL 187046"

Query Syntax:             PROGram[:SElected]:MAlLocate?

Return Format:            Integer

Attribute Summary:       Option: 1C2 Instrument BASIC  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

If you send PROG:MALL DEF, the analyzer resizes the stack space to fit the current active program. In some cases, the analyzer may allocate more memory than the Instrument BASIC program needs.

---

**Note**                    You need to allocate more memory if you encounter the message, "ERROR 2 Memory overflow" while your program is running.

---

---

**PROGram[:SElected]:NAME**

**command/query**

Selects an Instrument BASIC program.

Command Syntax:           PROGram [:SElected] :NAME  
                               PROGram1 | PROGram2 | PROGram3 | PROGram4 | PROGram5

Example Statements:        OUTPUT 711;"prog:selected:name PROGRAM4 "  
                               OUTPUT 711;"Program:Name PROGRAM1 "

Query Syntax:              PROGram [:SElected] :NAME?

Return Format:              PROG1 | PROG2 | PROG3 | PROG4 | PROG5

Attribute Summary:         Option:   1C2 Instrument BASIC  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: confirmed

Description:

Use this command to designate an Instrument BASIC program buffer as the “active” program buffer.

For example, use this command to select a program buffer when you load an Instrument BASIC program into the analyzer with the PROG:DEF command.



**PROGram[:SElected]:NUMBer****command/query**

Loads a new value for the specified numeric variable in the active Instrument BASIC program.

Command Syntax:           PROGram[:SElected]:NUMBer '<VARIABLE>', <BLOCK>

<VARIABLE> ::= name of a numeric variable

When data is ASCII-encoded (FORMat ASC), <BLOCK> takes the following form:

<BLOCK> ::= <number>[,<number>] . . .

<number> ::= a real number  
(NRf data)  
limits: - 9.9e37:9.9e37

When data is binary-encoded (FORMat REAL), <BLOCK> takes the following form:

<BLOCK> ::= #<byte>[<length\_bytes>]<number>[,<number>] . . .

<byte> ::= number of length bytes to follow (ASCII encoded)

<length\_bytes> ::= number of data bytes to follow (ASCII encoded)

<number> ::= a real number (32- or 64-bit binary floating point)

Example Statements:       OUTPUT 711;"program:sel:numb 'Address',11"  
                          OUTPUT 711;"PROG:NUMB 'Scode',7"

Query Syntax:            PROGram[:SElected]:NUMBer?

Return Format:            definite length <BLOCK>

Attribute Summary:       Option: 1C2 Instrument BASIC  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

When you load an array with this command, values in the <BLOCK> parameter are loaded into the 1st through nth elements of the array (where n is number of values in the block).

The analyzer uses the format specified by the FORMat[:DATA] command for query responses. The analyzer generates an error if the specified variable is not defined in the active program. Use the PROG:NAME command to specify the active program.

Use the PROG:STR command to load string variables.

---

**PROGram[:SElected]:STATe**

**command/query**

Selects the state of the active Instrument BASIC program.

Command Syntax:            PROGram[:SElected]:STATe STOP|PAUSE|RUN|CONTINUE

Example Statements:        OUTPUT 711;":PROG:STATE CONTINUE"  
                               OUTPUT 711;"prog:selected:stat RUN"

Query Syntax:              PROGram[:SElected]:STATe?

Return Format:              STOP|PAUS|RUN|CONT

Attribute Summary:        Option: 1C2 Instrument BASIC  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: confirmed

Description:

This command allows you to run, pause, stop or continue the active Instrument BASIC program.

The analyzer generates an error message, "Settings conflict; Invalid program state change requested," if you send RUN or CONT while a program is running. It also generates the error if you send CONT while a program is stopped.

Use the PROG:NAME command to select the active program.

**PROGram[:SElected]:STRing****command/query**

Loads a new value for the specified string variable for the active Instrument BASIC program.

Command Syntax:           PROGram[:SElected]:STRing '<VARIABLE>', '<STRING>'

<VARIABLE> ::= name of string variable  
(mandatory "\$" at the end of the name)

<STRING> ::= ASCII characters - 0 through 255  
maximum number of characters: 32766

Example Statements:       OUTPUT 711;"PROGRAM:SEL:STR 'A\$', 'Done'"  
                          OUTPUT 711;"prog:str 'Message\$', 'Measuring'"

Query Syntax:             PROGram[:SElected]:STRing?

Return Format:            "<STRING>"

Attribute Summary:       Option: 1C2 Instrument BASIC  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

This command sets or queries the contents of a string variable in the active Instrument BASIC program. Use the PROG:NAME to designate the active program.

Use the PROG[:SEL]:NUMB command to load or query numeric variables.



[SENSe:]

---



## [SENSe:]

---

Commands in this subsystem determine how measurement data is acquired.

Commands grouped under the AVERage keyword define how the results of several measurements will be combined into one trace.

Commands grouped under the DATA keyword allow uploading and downloading of time capture data.

Commands grouped under the FREQuency keyword control the frequency characteristics of the analyzer.

Commands grouped under the TCAPture keyword define the time capture parameters.

Commands grouped under the SWEEp keyword define parameters for swept sine measurements (Option 1D2).

Commands grouped under the VOLTage subsection control the amplitude characteristics of the input channels.

Commands under the WINDow keyword define windowing parameters.

SENSe is an implied mnemonic. Therefore, you can omit it from all SENSe commands. See “Implied Mnemonics” in chapter 2 of the GPIB Programmer’s Guide for more information.

[SENSe:]

---

**[SENSe:]AVERage:CONFidence**

**command/query**

Specifies the confidence level used in equal confidence averaging in octave measurements.

Command Syntax:            [SENSe:]AVERage:CONFidence {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: 0.25:2.0

<unit> ::= [DB]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"Average:Conf 0.25"  
                              OUTPUT 711;"SENS:AVER:CONF 2"

Query Syntax:             [SENSe:]AVERage:CONFidence?

Return Format:             Real

Attribute Summary:        Option: 1D1 Realtime Octave  
                              Synchronization Required: no  
                              Preset State: 0.5 DB  
                              SCPI Compliance: instrument-specific

Description:

The analyzer varies the average time constant to provide a 68% probability that the measured results are within  $\pm \sigma$  of the true mean value. There is a 95% probability that the results are within  $\pm 2 \sigma$  of the true value.

There are four values of  $\sigma$  available: .25 dB, .5 dB, 1 dB and 2 dB.



**[SENSe:]AVERAge:COUNT****command/query**

Specifies a count or a weighting factor for the averaged measurement data.

Command Syntax: [SENSe:]AVERAge:COUNT <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
 limits: 1:9999999  
 <step> ::= UP|DOWN  
 <bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"sense:average:count 2"  
 OUTPUT 711;"AVER:COUN 100"

Query Syntax: [SENSe:]AVERAge:COUNT?

Return Format: Integer

Attribute Summary: Option: not applicable  
 Synchronization Required: no  
 Preset State: +10  
 SCPI Compliance: confirmed

## Description:

As a counter, AVER:COUN determines the number of time records used to average the measurement data. This command is only valid in FFT, order analysis and correlation instrument modes.

Once the specified number of time records have been averaged, the No Pending Operation (NPO) flag is set to 1. Use the \*OPC command to determine when the specified number of time records have been combined. (See "Synchronization" in the GPIB Programmer's Guide for more information about the completion of averaged measurements.)

When used with exponential averaging (AVER:TYPE RMS;TCON EXP or AVER:TYPE TIME;TCON EXP), this command determines how the results of the current measurement (new data) is combined with the averaged trace (old data). Data is combined, point by point, according to the following formula:

$$\frac{1}{n} \text{new} + \frac{n-1}{n} \text{old}$$

[SENSe:]

---

**[SENSe:]AVERAge:HOLD**

**command/query**

Specifies the type of hold used in averaging octave measurements.

Command Syntax:            [SENSe:]AVERAge:HOLD OFF|0|MAXimum|MINimum

Example Statements:        OUTPUT 711;"sens:average:hold MINIMUM"  
                              OUTPUT 711;"Aver:Hold OFF"

Query Syntax:              [SENSe:]AVERAge:HOLD?

Return Format:              OFF|MAX|MIN

Attribute Summary:        Option: 1D1 Realtime Octave  
                              Synchronization Required: no  
                              Preset State: OFF  
                              SCPI Compliance: instrument-specific

Description:

This command specifies the type of average hold used for octave measurements. It is valid when used with:

- Linear averaging (AVER:TYPE RMS).
- Exponential averaging (AVER:TYPE RMS;TCON EXP).
- Equal confidence averaging (AVER:TYPE ECON).

If you send AVER:HOLD MAX, the analyzer displays the maximum averaged spectrum value for each band.

If you send AVER:HOLD MIN, the analyzer displays the minimum averaged spectrum value for each band. This is useful for estimating background noise.

---

**Note**                      AVERAge:HOLD differs from the peak hold function. See the AVER:TYPE MAX command description for information about the peak hold function.

---

**[SENSe:]AVERage:IMPulse****command/query**

Enables impulse detection in octave measurements.

Command Syntax:           [SENSe:]AVERage:IMPulse OFF|0|ON|1

Example Statements:        OUTPUT 711;":AVER:IMPULSE ON"  
                              OUTPUT 711;"sens:aver:impulse ON"

Query Syntax:             [SENSe:]AVERage:IMPulse?

Return Format:             Integer

Attribute Summary:        Option: 1D1 Realtime Octave  
                              Synchronization Required: no  
                              Preset State: OFF (+0)  
                              SCPI Compliance: instrument-specific

**Description:**

When impulse detection is on (AVER:IMP ON), the analyzer computes and displays the IEC 651 impulse characteristics in the overall power band.

In linear averaging (AVER:TYPE RMS;TCON FRE), the analyzer calculates the value of the impulse output over the average time (specified with the AVER:TIME command).

In all other types of averaging, the analyzer calculates the instantaneous value of the impulse vector.

See the analyzer's Online Help for additional information.

[SENSe:]

---

**[SENSe:]AVERage:IREsult:RATE**

**command/query**

Specifies how often the display is updated when fast average mode is on.

Command Syntax: [SENSe:]AVERage:IREsult:RATE <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 1:9999999  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"Sens:Average:Ires:Rate 8575803"  
                      OUTPUT 711;"AVERAGE:IRES:RATE 3430970"

Query Syntax: [SENSe:]AVERage:IREsult:RATE?

Return Format: Integer

Attribute Summary: Option: not applicable  
                      Synchronization Required: no  
                      Preset State: +5  
                      SCPI Compliance: instrument-specific

Description:

This command specifies the rate used by the analyzer when AVER:IRES is on.

The analyzer updates the display once for each N averages. N is the update rate specified with this command. The analyzer continues to update the display whenever it reaches a multiple of N.

**[SENSe:]AVERage:IREsult[:STATe]****command/query**

Selects fast average mode.

Command Syntax: [SENSe:]AVERage:IREsult[:STATe] OFF|0|ON|1

Example Statements: OUTPUT 711;":aver:ires ON"  
OUTPUT 711;":Sense:Aver:Iresult:Stat OFF"

Query Syntax: [SENSe:]AVERage:IREsult[:STATe]?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF (+0)  
SCPI Compliance: instrument-specific

## Description:

This command specifies whether the analyzer displays in fast average mode.

In FFT and correlation instrument mode, AVER:IRES ON updates the display once for every N averages. N is the update rate specified with AVER:IRES:RATE. The preset AVER:IRES rate is 5. OFF updates the display after each average.

In swept sine instrument mode, AVER:IRES ON updates the display after the entire sweep is completed. OFF updates the display at each point in the sweep.

In histogram instrument mode, AVER:IRES ON updates the display as fast as it can without slowing down the measurement. OFF updates the display at the end of the measurement.

[SENSe:]

---

**[SENSe:]AVERage:PREView**

**command/query**

Specifies the type of preview averaging.

Command Syntax: [SENSe:]AVERage:PREView OFF|0|MANual|TIMed

Example Statements: OUTPUT 711;"SENS:AVERAGE:PREV TIMED"  
OUTPUT 711;"average:prev TIMED"

Query Syntax: [SENSe:]AVERage:PREView?

Return Format: OFF|MAN|TIM

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF  
SCPI Compliance: instrument-specific

Description:

This command is valid in FFT analysis mode only (INST:SEL FFT).

To enable manual previewing, send AVER:PREV MAN. The analyzer waits for a response before taking the next time record.

To enable timed previewing, send AVER:PREV TIM. The analyzer waits for a specified amount of time before accepting the time record. That is, if no response is sent the analyzer accepts the time record. Set the time period with the AVER:PREV:TIME command.

After each time record is collected, the Waiting for Accept/Reject bit in the Operation Status Register is set to 1. The bit is cleared when the analyzer receives an accept or reject command or when the analyzer receives a command that changes the measurement setup.

To accept the time record send AVER:PREV:ACC. To reject the time record send AVER:PREV:REJ.  
To turn off average previewing, send AVER:PREV OFF.

---

**[SENSe:]AVERage:PREView:ACcept****command**

Accept the current time record during preview averaging.

Command Syntax:           [SENSe:]AVERage:PREView:ACcept

Example Statements:        OUTPUT 711;":Aver:Preview:Acc"  
                              OUTPUT 711;"SENSE:AVER:PREV:ACCEPT"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

[SENSe:]

---

**[SENSe:]AVERage:PREView:REJect**

**command**

Reject the current time record during preview averaging.

Command Syntax:            [SENSe:]AVERage:PREView:REJect

Example Statements:        OUTPUT 711;"sens:average:prev:rej"  
                              OUTPUT 711;"Average:Prev:Reject"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific



**[SENSe:]AVERage:PREView:TIME****command/query**

Specifies the amount of time the analyzer waits for a response in timed preview averaging.

Command Syntax: [SENSe:]AVERage:PREView:TIME {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: 0.1:3600.0

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;":aver:prev:time 20"  
OUTPUT 711;"SENS:AVERAGE:PREVIEW:TIME 60 S"

Query Syntax: [SENSe:]AVERage:PREView:TIME?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: 10 S  
SCPI Compliance: instrument-specific

Description:

The analyzer waits the specified amount of time for a response. If a response is not sent, the analyzer accepts the time record.

Time is specified in seconds. Specify timed preview averaging with the AVER:PREV TIM command.

[SENSe:]

---

**[SENSe:]AVERage[:STATe]**

**command/query**

Turns the selected averaging function (AVER:TYPE) on or off.

Command Syntax: [SENSe:]AVERage[:STATe] OFF|0|ON|1

Example Statements: OUTPUT 711;":AVER OFF"  
OUTPUT 711;"sens:average:stat ON"

Query Syntax: [SENSe:]AVERage[:STATe]?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF (+0)  
SCPI Compliance: confirmed

**Description:**

When you select ON, each trace represents the combined results of several measurements, and the averaging function specified in AVER:TYPE determines how results are combined.

RMS averaging (AVER:TYPE RMS) provides a better estimate of the noise in measurement data. Vector averaging (AVER:TYPE TIME) reduces the amount of random noise and provides a better estimate of the repetitive signals in the measurement data. Maximum averaging (AVER:TYPE MAX) saves the maximum power value (power spectra) for each frequency bin.

When you select OFF, each trace represents the results of a single measurement. It is mathematically equivalent to exponential RMS (power) averaging with 1 average.

When averaging is ON and AVER:TYPE is MAXimum, RMS or TIME, INIT:IMM sets the No Pending Operation (NPO) flag to 1 after the specified number (set with the AVER:COUN command) of measurement results have been combined. When averaging is ON and termination control is exponential (AVER:TCON EXPO), INIT:IMM sets the NPO flag to 1 each time a measurement is completed, after the initial N averages. When averaging is OFF, INIT:IMM sets the NPO flag to 1 each time a measurement is completed. It acts as if the average count is set to 1. See "Synchronization" in the GPIB Programmer's Guide for more information about the completion of averaged measurements.

Figure 18-1 illustrates of transition of bits (Measuring, Averaging, Waiting for TRIG, and Waiting for ARM) in the Operation Status register.

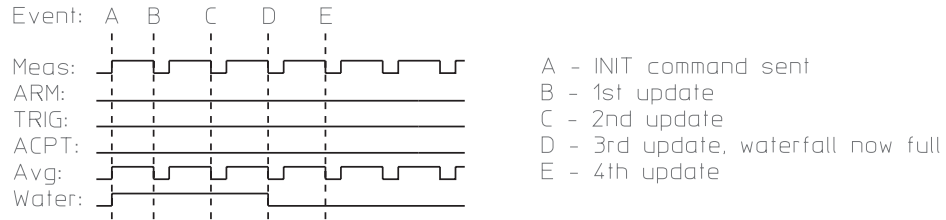
---

**Note** Trigger conditions must be met for each measurement—even when averaging is turned on.

---

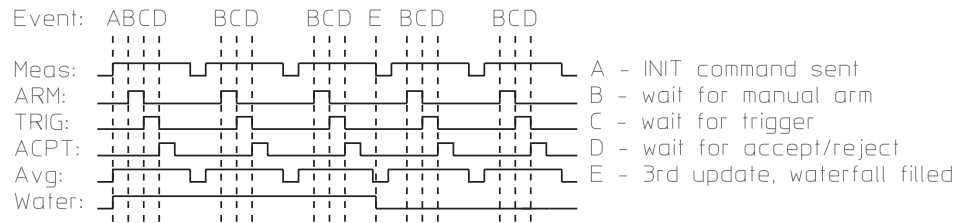
**Instrument Setting: PRESET state**

ARM:SOURce IMMEDIATE  
 AVERAge ON  
 AVERAge:IREsult: RATE 4  
 CALCulate:WATerfall: COUNt 3  
 TRIGger:SOURce IMMEDIATE



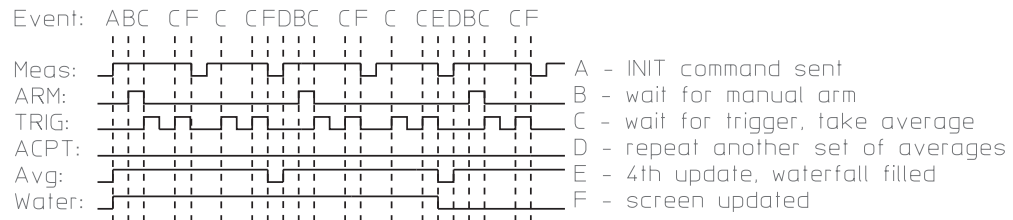
**Instrument Setting: Manual arm, triggered, preview on**

ARM:SOUR MAN  
 AVER:PREV MAN  
 CALC:WAT:COUN 3  
 TRIG:SOUR INT1



**Instrument Setting: Fast average, repeat on, manual arm, input trigger**

ARM:SOUR MAN                      AVER:IREs:RATE 2  
 AVER ON                              CALC:WAT:COUN 4  
 AVER:COUN 4                        TRIG:SOUR INT1  
 AVER:TCON REP  
 AVER:IREs OFF



**Figure 18-1. Transition of Operation Status Register Bits When AVERage[:STATe] ON.**

[SENSe:]

---

**[SENSe:]AVERAge:TCONtrol**

**command/query**

Specifies how the analyzer behaves after the count (AVER:COUN) is reached.

Command Syntax:            [SENSe:]AVERAge:TCONtrol FREeze|REPeat|EXPonential

Example Statements:        OUTPUT 711;"Sense:Aver:Tcon FREEZE"  
                              OUTPUT 711;"AVERAGE:TCON REPEAT"

Query Syntax:              [SENSe:]AVERAge:TCONtrol?

Return Format:             FRE|REP|EXP

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This command specifies termination control during averaging.

To specify linear (normal) averaging, send TCON FRE with the AVER:TYPE command.

To specify exponential averaging, send TCON EXP with the AVER:TYPE command.

To specify repeat averaging ("autostart"), send TCON REP with the AVER:TYPE command. The analyzer takes N averages, clears the data, waits for arming conditions, and then takes another N averages. The analyzer continues taking measurements until you send one of the following commands:

- INIT:CONT OFF
- AVER:STAT OFF
- AVER:TCON FRE

Depending upon the instrument mode (specified with the INST:SEL command), some types of termination control and averaging are not valid. See table 18-1.

Table 18-1. Valid Types of Termination Control and Averaging for Instrument Mode

## FFT instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	valid	valid	not allowed
RMS	valid	valid	valid
TIME	valid	valid	valid
ECONfidence	not allowed	not allowed	not allowed

## Correlation instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	not allowed	not allowed	not allowed
RMS	valid	valid	valid
TIME	valid	valid	valid
ECONfidence	not allowed	not allowed	not allowed

## Histogram instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	not allowed	not allowed	not allowed
RMS	not allowed	not allowed	not allowed
TIME	valid	valid	not allowed
ECONfidence	not allowed	not allowed	not allowed

## Octave instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	valid	valid	not allowed
RMS	valid	valid	valid
TIME	not allowed	not allowed	not allowed
ECONfidence	not allowed	not allowed	valid

[SENSe:]

Order instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	not allowed	not allowed	not allowed
RMS	not allowed	not allowed	not allowed
TIME (time)	valid	valid	valid
ECONfidence	not allowed	not allowed	not allowed

Swept Sine instrument mode

AVERage:TYPE	AVERage:TCONTROL		
	FREeze	REPeat	EXPonential
MAX	not allowed	not allowed	not allowed
RMS	not allowed	not allowed	not allowed
TIME	not allowed	not allowed	not allowed
ECONfidence	not allowed	not allowed	not allowed

**[SENSe:]AVERage:TIME****command/query**

Specify the time period used in averaging octave measurements and histograms.

Command Syntax: [SENSe:]AVERage:TIME {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: 0.0:9.9e37

<unit> ::= [S|REC|PNT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;":average:time 3.77454e+36"  
OUTPUT 711;"Sens:Average:Time 3.72708e+37"

Query Syntax: [SENSe:]AVERage:TIME?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: .125 S (INST:SEL OCT)  
1 REC (INST:SEL HIST)  
SCPI Compliance: instrument-specific

Description:

---

**Note** This command is only valid in histogram instrument mode (INST:SEL HIST) and octave analysis instrument mode (INST:SEL OCT; Option 1D1).

---

In octave analysis instrument mode (INST:SEL OCT) :

The specified amount of time (in seconds) is used in linear and exponential averaging. It is also used in the peak hold function.

In linear averaging (AVER:TYPE RMS), the value is used for linear integration time. In exponential averaging (AVER:TYPE RMS;TCON EXP), the value is used as the time constant.

In peak hold (AVER:TYPE MAX), this value is used as the integration time over which to hold maximum values. The termination control (AVER:TCON) must be FREeze.

[SENSe:]

In histogram analysis instrument mode (INST:SEL HIST) :

This command specifies the length of time averaging data for the histogram. This histogram length can be specified in time (S for seconds), records (REC), or points (PNT). The analyzer rounds the specified histogram length up to the nearest point.

An optimal histogram may be obtained by setting the number of points (specified by this command) to the number of bins<sup>2</sup> (specified with the HIST:BINS command).

$$\backslash \textit{Optimal Histogram} \approx (\textit{AVER:TIME}) = (\textit{HIST:BINS})^2$$



**[SENSe:]AVERage:TYPE****command/query**

Specifies the type of averaging the analyzer performs.

Command Syntax: [SENSe:]AVERage:TYPE MAXimum|RMS|TIME|VECTor|ECONfidence

Example Statements: OUTPUT 711;"SENSE:AVER:TYPE TIME"  
OUTPUT 711;"average:type RMS"

Query Syntax: [SENSe:]AVERage:TYPE?

Return Format: MAX|RMS|TIME|ECON

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: RMS  
SCPI Compliance: confirmed

Description:

The types of averaging available vary according to the instrument mode.

FFT analysis instrument mode (INST:SEL FFT) :

To select rms (power) averaging, send AVER:TYPE RMS. The analyzer averages N time records where N is the number of averages you specify with the AVER:COUN command.

To select exponential rms (power) averaging, send AVER:TYPE RMS;TCON EXP. The number of averages specified with AVER:COUN determines the weighting of old versus new data.

To select time averaging, send AVER:TYPE TIME. The analyzer averages complex values point-by-point in the frequency domain. The averaged frequency domain spectra is transformed (inverse FFT) to give averaged time data. The input signal must be periodic and a trigger signal from the analyzer's source or from an external signal must be provided.

To select exponential time averaging, send AVER:TYPE TIME;TCON EXP. The number of averages specified with AVER:COUN determines the weighting of old versus new data.

Send AVER:TYPE MAX to select the peak hold function. The analyzer takes data continuously and mathematically compares each data point along the measured frequency span with the previous peak values. Only the largest value of each point is saved. The results are not mathematically averaged.

[SENSe:]

In correlation analysis instrument mode (INST:SEL CORR), the following commands are valid:

AVER:TYPE RMS

AVER:TYPE RMS;TCON EXP

AVER:TYPE TIME

AVER:TYPE TIME;TCON EXP

In histogram instrument mode (INST:SEL HIST) :

This command is not valid.

Octave analysis instrument mode (INST:SEL OCT) :

To select linear averaging, send AVER:TYPE RMS. Old and new data records are weighted equally to yield the arithmetic mean. Averaging is done for a specified amount of time rather than for a number of averages. The value specified with the AVER:TIME command is used for linear integration time.

To select exponential averaging, send AVER:TYPE RMS;TCON EXP. New data is weighted more than old data. The AVER:TIME command specifies the time constant.

To select equal confidence averaging, send AVER:TYPE ECON;TCON EXP. The averaging time for each band is proportional to the bandwidth product for that band. The relative confidence in the measurement is equal across bands. The AVER:CONF command determines the specifies the confidence level. This is an instrument-specific SCPI command.

To specify the peak hold function, send AVER:TYPE MAX. The value specified with the AVER:TIME command is the integration time over which to hold maximum values.

In swept sine instrument mode (INST:SEL SINE) :

This command is not valid. Averaging integrates a single data point at a time, as opposed to averaging complete time records.

**[SENSe:]DATA****command/query**

Uploads or downloads time-capture data between the analyzer and the controller.

Command Syntax: [SENSe:]DATA TCAP1 | TCAP2 | TCAP3 | TCAP4, <DATA>

```
<DATA> ::= <DEF_BLOCK>
        ::= <Nrf>, <Nrf>, <Nrf>, ...
```

Example Statements: OUTPUT 711;"SENSE:DATA? TCAP1"  
(See example program "GETCAP" in appendix F.)

Query Syntax: [SENSe:]DATA? TCAP1 | TCAP2 | TCAP3 | TCAP4

Return Format: DEF\_USER

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset state: not applicable  
SCPI Compliance: instrument-specific

## Description:

The command takes one parameter and a block of data. The parameter specifies the channel of the time capture buffer that the data will be loaded into, TCAP1 for channel 1, TCAP2 for channel 2, TCAP3 for channel 3, TCAP4 for channel 4.

The analyzer can only load definite length blocks; it cannot load indefinite blocks or ASCII data. The data is stored internally as 16 bit integers, but it is transferred as floating point values. The floating-point numbers are scaled and converted to integers with the time capture range value specified with the [SENSe:]DATA:RANGE TCAP[1|2|3|4] command. Using the time-capture range to scale the data yields the best dynamic range. The number of points sent to the analyzer must be the same as the number of points in the capture buffer.

The query form of this command is used to transfer time-capture data to the controller. The TCAP1|TCAP2|TCAP3|TCAP4 parameter specifies the channel. The data is transferred to the controller in the format specified by the FORM:DATA command.

The time-capture buffer start and stop frequency can be used to calculate the time spacing between the points. Use the SENSE:DATA:HEAD:FREQ:START and SENSE:DATA:HEAD:FREQ:STOP commands to determine the start and stop frequencies.

Use the SENSE:TCAP:FILE command if you want to upload and download the time-capture file.

[SENSe:]

---

**[SENSe:]DATA:HEADer:FREQuency:STARt?**

**query**

Returns the start frequency setting used for the current time-capture buffer.

Query Syntax:                    [SENSe:]DATA:HEADer:FREQuency:STARt?

Example Statements:            OUTPUT 711;"SENSe:DATA:HEAD:FREQ:STARt?"

Return Format:                  Real  
Attribute Summary:  
Option: not applicable  
Synchronization Required: no  
Preset state: 0  
SCPI Compliance: instrument-specific

Description:

This command is used to determine the start frequency used when the data in the current time-capture buffer was captured.

If the value returned by this query is non-zero, the capture data returned by the SENSE:DATA? query will be complex.

The time-capture start and stop frequency can be used to calculate the time spacing between the capture data points. The spacing is equal to  $400 / ((\text{Stop frequency}) - (\text{Start frequency}))$ .

---

**[SENSE:]DATA:HEADer:FREQuency:STOP?****query**

Returns the stop frequency setting used for the current time-capture buffer.

Query Syntax:                    [SENSE:]DATA:HEADer:FREQuency:STOP?

Example Statements:            OUTPUT 711;"SENSE:DATA:HEAD:FREQ:STOP?"

Return Format:                    Real

Attribute Summary:            Option: not applicable  
                                 Synchronization Required: no  
                                 Preset state: 0  
                                 SCPI Compliance: instrument-specific

Description:

This command is used to determine the stop frequency used when the data in the current time-capture buffer was captured.

The time-capture start and stop frequency can be used to calculate the time spacing between the capture data points. The spacing is equal to  $400 / ((\text{Stop frequency}) - (\text{Start frequency}))$ .

[SENSe:]

---

**[SENSe:]DATA:HEADer:POINts**

**query**

Returns the number of points for the time capture buffer specified.

Query Syntax: [SENSe:]DATA:HEADer:POINts? TCAP1 | TCAP2 | TCAP3 | TCAP4

Example Statements: OUTPUT 711;"SENSe:DATA:HEAD:POIN? TCAP1"

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset state: 0  
SCPI Compliance: instrument-specific

Description:

This query is used to determine the number of data points in the specified time-capture buffer. This information may be required to transfer the contents of the buffer from the analyzer to the controller.

If the time-capture was performed in zoom mode, the time-capture data will be complex. The number of points transferred by the analyzer with the SENSE:DATA? query will then be twice as many as the value returned by this query. Use the SENSE:DATA:HEAD:FREQ:STAR? query to determine if the capture was performed in zoom mode.

---

**[SENSe:]DATA:RANGe****command/query**

Specifies range value for the time capture data.

Command Syntax: [SENSe:]DATA:RANGe TCAP1|TCAP2|TCAP3|TCAP4, <number>

<number> ::= a real number (NRf data)

limits: 1.0E-20 - 1.0E20

Example Statements: OUTPUT 711;"SENSe:DATA:RANGe TCAP1, 1.5e+3"

Query Syntax: [SENSe:]DATA:RANGe? TCAP1|TCAP2|TCAP3|TCAP4

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset state: 0  
SCPI Compliance: instrument-specific

**Description:**

This command is used to set the range value for data transferred to the analyzer with the SENSE:DATA command. The range must be set before time data is loaded into the capture buffers from a controller. This value (in Volts or EUs) is used to convert the data into the integer format used by the analyzer. To get full 16 bit accuracy, the value sent should be the same as the largest data value to be loaded.

---

**Note:25** Any data values larger than the range value will be clipped.

---

[SENSe:]

---

**[SENSe:]FEED**

**command/query**

Specifies the data source for a measurement; either from the input channels or from the time capture buffer.

Command Syntax:            [SENSe:]FEED INPut|TCAPture

Example Statements:        OUTPUT 711;":Feed TCAPTURE"  
                              OUTPUT 711;"SENS:FEED TCAPTURE"

Query Syntax:              [SENSe:]FEED?

Return Format:              INP|TCAP

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: INP  
                              SCPI Compliance: instrument-specific

Description:

Data from the time capture buffer may be used for a measurement, replacing the use of the inputs (channels 1, 2, 3, and 4). This command directs the analyzer to use the data in the time capture buffer for the measurement.

---

**Note**                      The measurement does not begin until the INITiate command is sent.

---



**[SENSe:]FREQUency:BLOCKsize****command/query**

Specifies the number of real-time data points displayed on the analyzer's screen.

Command Syntax: [SENSe:]FREQUency:BLOCKsize <number>|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 256:2048
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements: OUTPUT 711;":FREQUENCY:BLOCK 512"  
OUTPUT 711;"sens:freq:bloc 1024"

Query Syntax: [SENSe:]FREQUency:BLOCKsize?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

**Description:**

This command sets the number of points in a time record for FFT and correlation instrument modes.

In FFT analysis instrument mode (INST:SEL FFT) the number of frequency-data points depends upon the blocksize. See table 18-2. This command is similar to the FREQUency:RESolution command which specifies the resolution with frequency lines.

In correlation analysis instrument mode (INST:SEL CORR) the number of displayed time-data points depends upon the blocksize. See table 18-2.

[SENSe:]

Table 18-2. Number of displayed data points according to blocksize  
 FFT Instrument Mode

Blocksize Number of real time points (time data)	Baseband	Zoom (Start frequency $\sqrt{0}$ )	
	Number of frequency points (frequency data)	Number of frequency points (frequency data)	Number of complex time points (real/ imaginary pairs) (time data)
256	101	101	128
512	201	201	256
1024	401	401	512
2048	801	801	1024

Correlation Instrument Mode  
 (no complex data)

Blocksize Time domain (real data)	Auto- and cross correlation (real data)		
	0 to T/2	-T/2 to T/2	-T/4 to T/4
256	128	256	128
512	256	512	256
1024	512	1024	512
2048	1024	2048	1024

**[SENSe:]FREQUency:CENTer****command/query**

Specifies the center frequency for the current measurement.

Command Syntax:           [SENSe:]FREQUency:CENTer {<number> [<unit>] } | <step> | <bound>

```
<number> ::= a real number (NRf data)
           limits: 0.0234375:115000.0
<unit>   ::= [HZ|CPM]
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements:       OUTPUT 711;"sense:freq:center 93890.6"  
                          OUTPUT 711;"Freq:Cent 51351.5"

Query Syntax:             [SENSe:]FREQUency:CENTer?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +5.12E+004  
                          SCPI Compliance: confirmed

Description:

FREQ:CENT and FREQ:SPAN work together to define the band of frequency you want to analyze. When you change the value of FREQ:CENT, the value of FREQ:SPAN remains constant.

Step size (FREQ:STEP) determines the change in frequency which results when you send UP or DOWN with this command.

In swept sine instrument mode (INST:SEL SINE; Option 1D2):

The allowable values are 15.625 mHz to 511199.984375 Hz in 2 channel instrument mode and 31.25 mHz to 25599.984375 Hz in 4 channel instrument mode. A value specified by this command is rounded to the next lower 15.625 mHz step.

[SENSe:]

---

**[SENSe:]FREQUency:MANual**

**command/query**

Selects a discrete point to be measured during manual sweep mode.

Command Syntax: [SENSe:]FREQUency:MANual {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: 0.015625:51200.0

<unit> ::= [HZ]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;":FREQUENCY:MAN 1388.13"  
OUTPUT 711;"sense:freq:man 24158.9"

Query Syntax: [SENSe:]FREQUency:MANual?

Return Format: Real

Attribute Summary: Option: 1D2 Swept Sine  
Synchronization Required: no  
Preset State: 51.2 HZ  
SCPI Compliance: confirmed

Description:

The frequency must fall within the start and stop frequencies (FREQ:STAR and FREQ:STOP). Multiple points can be measured by repeating this command with each value.

---

**Note** This command is only used if the SWE:MODE MAN command has been sent.

---



[SENSe:]

Table 18-3. Number of displayed data points in variable resolution

FFT Instrument Mode

Resolution	Baseband		Zoom (Start frequency $\sqrt{0}$ )	
	Number of frequency points (frequency data)	Number of real time points (time data)	Number of frequency points (frequency data)	Number of complex time points (real/ imaginary pairs) (time data)
100	101	256	101	128
200	201	512	201	256
400	401	1024	401	512
800	801	2048	801	1024

Swept Sine Instrument Mode  
(FREQ:RES:AUTO ON)

FREQ:RES:AUTO:MIN (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to FREQ:SPAN / FREQ:RES:AUTO:MIN. With PCT, the spacing between measurement points is a percentage of the total frequency span.

In swept sine instrument mode (INST:SEL SINE; Option 1D2):

If linear spacing is specified (SWE:SPAC LIN), the resolution can be set using the following units:

- HZ hertz
- PNT/SWP points per sweep
- PCT spacing between measurement points as a percentage of the total frequency span

If logarithmic spacing is specified (SWE:SPAC LOG), the resolution can be set using the following units:

- PNT/SWP points per sweep
- PNT/DEC points per decade
- PNT/OCT points per octave
- PCT spacing between measurement points as a percentage of the total frequency span

To determine what the current unit setting is, send `FREQ:RES? UNIT`. See table 18-4 to determine the number of displayed data points with swept sine variable resolution.

---

**Note** This command is not used in swept sine if the `FREQ:RES:AUTO ON` command is sent.

---

Table 18-4. Number of displayed data points in variable resolution

Swept Sine Instrument Mode  
(FREQ:RES:AUTO OFF)

FREQ:RES (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to  $\text{FREQ:SPAN} / \text{FREQ:RES (PNT/SWP)}$ . With PCT, the spacing between measurement points is a percentage of the total frequency span.

[SENSe:]

---

**[SENSe:]FREQuency:RESolution:AUTO**

**command/query**

Selects auto resolution for swept sine instrument mode.

Command Syntax:            [SENSe:]FREQuency:RESolution:AUTO OFF|0|ON|1

Example Statements:        OUTPUT 711;"Sense:Freq:Resolution:Auto OFF"  
                              OUTPUT 711;"FREQ:RESOLUTION:AUTO OFF"

Query Syntax:              [SENSe:]FREQuency:RESolution:AUTO?

Return Format:              Integer

Attribute Summary:        Option: 1D2 Swept Sine  
                              Synchronization Required: no  
                              Preset State: +0  
                              SCPI Compliance: confirmed

Description:

The frequency spacing between measurement points is adjusted automatically by the analyzer. The analyzer increments or decrements the size of the step to accommodate varying frequency response changes.

The analyzer calculates the ratio of the frequency response of the current point to the frequency response of the previous point. If the ratio exceeds the maximum percentage change specified by the FREQ:RES:AUTO:MCH command, the analyzer adjusts the resolution to measure the next point.

The adjusted resolution value is never less than the minimum resolution value specified by the FREQ:RES:AUTO:MIN command. The initial resolution of the sweep between the first two points is specified by FREQ:RES:AUTO:MIN.

---

**Note**                      FREQ:RES:AUTO is ON after a reset (\*RST).

---

See table 18-5 to determine the number of displayed data points with swept sine variable resolution.



Table 18-5. Number of displayed data points in swept sine variable resolution

Swept Sine Instrument Mode  
(FREQ:RES:AUTO ON)

FREQ:RES:AUTO:MIN (Number of measured points <sup>1</sup> )	Number of displayed points
3 to 401 PNT/SWP	401
402 to 801 PNT/SWP	801

<sup>1</sup> The spacing of measurement points in hertz is equal to FREQ:SPAN / FREQ:RES:AUTO:MIN. With PCT, the spacing between measurement points is a percentage of the total frequency span.

[SENSe:]

---

**[SENSe:]FREQuency:RESolution:AUTO:MCHange**

**command/query**

Specifies the maximum change permitted between the frequency response of the current measurement point and the frequency response of the previous measurement point.

Command Syntax:           [SENSe:]FREQuency:RESolution:AUTO:MCHange {<number>  
                          [<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0.00391:100

<unit> ::= [PCT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"sense:freq:resolution:auto:mch 30 PCT"  
                          OUTPUT 711;"FREQ:RES:AUTO:MCH .25"

Query Syntax:            [SENSe:]FREQuency:RESolution:AUTO:MCHange?

Return Format:           Real

Attribute Summary:      Option: 1D2 Swept Sine  
                          Synchronization Required: no  
                          Preset State: +2 PCT  
                          SCPI Compliance: instrument-specific

Description:

This command is used with auto resolution (FREQ:RES:AUTO ON).

The analyzer calculates the ratio of the frequency response of the current point to the frequency response of the previous point. The ratio exceeds the value specified by this command, the analyzer estimates a correction for the resolution and applies it to the next measurement point.

**[SENSe:]FREQuency:RESolution:AUTO:MINimum****command/query**

Specifies the initial resolution of a swept sine measurement with automatic resolution.

Command Syntax:           [SENSe:]FREQuency:RESolution:AUTO:MINimum {<number>  
                                  [<unit>]}|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 0.015625:51200.0
<unit>   ::= [HZ|PCT|PNT/SWP|PNT/DEC|PNT/OCT]
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements:       OUTPUT 711;"sense:freq:resolution:auto:minimum 10 hz"  
                                  OUTPUT 711;"FREQ:RES:AUTO:MIN 200 PNT/SWP"

Query Syntax:            [SENSe:]FREQuency:RESolution:AUTO:MINimum?

Return Format:            Real

Attribute Summary:       Option: 1D2 Swept Sine  
                                  Synchronization Required: no  
                                  Preset State: 401 PNT/SWP  
                                  SCPI Compliance: instrument-specific

## Description:

If automatic resolution is specified (FREQ:RES:AUTO ON), this command sets the initial resolution used between the first and second measurement point. It also specifies the minimum resolution the analyzer uses if an adjustment in resolution is required.

The analyzer calculates the ratio of the frequency response of the current measurement point to the frequency response of the previous measurement point. If the ratio exceeds the specified limit (FREQ:RES:AUTO:MCH), the analyzer corrects the resolution and applies it to the next measurement point. The correction is never less than the value specified by this command.

If linear spacing is specified (SWE:SPAC LIN), the resolution can be set using the following units:

- HZ hertz
- PNT/SWP points per sweep
- PCT spacing between measurement points as a percentage of the total frequency span

If logarithmic spacing is specified (SWE:SPAC LOG), the resolution can be set using the following units:

- PNT/SWP points per sweep
- PNT/DEC points per decade
- PNT/OCT points per octave
- PCT spacing between measurement points as a percentage of the total frequency span

To determine what the current unit setting is, send FREQ:RES? UNIT.

[SENSe:]

---

**[SENSe:]FREQUency:RESolution:OCTave**

**command/query**

Specifies the type of octave measurement.

Command Syntax: [SENSe:]FREQUency:RESolution:OCTave THIRd|FULL|TWELfth

Example Statements: OUTPUT 711;":frequency:res:oct THIRd"  
OUTPUT 711;"Sense:Freq:Resolution:Oct THIRd"

Query Syntax: [SENSe:]FREQUency:RESolution:OCTave?

Return Format: THIR|FULL|TWEL

Attribute Summary: Option: 1D0 Computed Order Tracking  
Synchronization Required: no  
Preset State: THIRd  
SCPI Compliance: instrument-specific

**Description:**

To select 1/1 octave band measurements, send `FREQ:RES:OCT FULL`. The center frequency of any band is twice the center frequency of the previous band. The analyzer displays a minimum of 1 frequency band, a maximum of 12 frequency bands, a weighted overall band and an overall band. See table 18-6 for the minimum start frequency and the maximum stop frequency.

To select 1/3 octave band measurements, send `FREQ:RES:OCT THIR`. The center frequency of each 1/3 octave frequency band is located at a frequency of  $2^{(1/3)}$  times the preceding 1/3 octave band. The analyzer displays a minimum of 3 frequency bands, a maximum of 33 frequency bands, a weighted overall band, and an overall band. See table 18-6 for the minimum start frequency and the maximum stop frequency.

To select 1/12 octave band measurements, send `FREQ:RES:OCT TWEL`. The center frequency of each 1/12 octave frequency band is located at a frequency of  $2^{(1/12)}$  times the preceding 1/12 octave band. The analyzer displays a minimum of 12 frequency bands, a maximum of 144 frequency bands, a weighted overall band, and an overall band. See table 18-6 for the minimum start frequency and the maximum stop frequency.

The analyzer maintains a minimum of one octave and a maximum of twelve octaves. It may decrease the start frequency or increase the stop frequency to meet this constraint. To specify start and stop frequencies, send the `FREQ:STAR` and `FREQ:STOP` commands.

Table 18-6. Octave Analysis  
 Range of Available Start and Stop Frequencies  
 (in hertz)

	1 Channel (INPut2 OFF)	2 Channel (INPut2 ON;INPut4 OFF)	4 Channel (INPut4 ON)
1/1 Octave	63 mHz to 16 kHz	63mHz to 8 kHz	63 mHz - 4 kHz
1/3 Octave	100 mHz to 40 kHz	100 mHz to 20 kHz	100 mHz to 10 kHz
1/12 Octave	99.7 mHz to 12.3 kHz	99.7 mHz to 6.17 kHz	99.7 mHz to 3.08kHz

[SENSe:]

---

**[SENSe:]FREQuency:SPAN**

**command/query**

Specifies the frequency bandwidth to be measured.

Command Syntax: [SENSe:]FREQuency:SPAN {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)

limits: 0.015625:102400.0

<unit> ::= [HZ|CPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;" :FREQ:SPAN 5E4 "  
OUTPUT 711;"sens:freq:span MIN"

Query Syntax: [SENSe:]FREQuency:SPAN?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +1.024E+005  
SCPI Compliance: confirmed

Description:

In FFT analysis instrument mode (INST:SEL FFT) :

The value of FREQ:SPAN is used with either FREQ:CENT or FREQ:STAR to define the band of frequencies. The maximum and the minimum frequency spans are listed in table 18-7. Allowable values for the frequency span are determined by the following formula:

$$(\text{maximum frequency span}) / 2^n$$

$$\text{where } 0 \leq n \leq 19$$

When you send this command, the value of the record length (SWE:TIME) is adjusted so the following formula is true:

$$\text{SWE:TIME} = \text{FREQ:RES} / \text{FREQ:SPAN}$$

The frequency span limits the range of the start frequency values (FREQ:STAR) according to the following formulas:

for one channel measurements:

$$(\text{FREQ:STAR}) \text{ @ } 115 \text{ kHz} - (\text{FREQ:SPAN}) / 2$$

for two channel measurements:

$$(\text{FREQ:STAR}) \text{ @ } 57.5 \text{ kHz} - (\text{FREQ:SPAN}) / 2$$

for four channel measurements:

$$(\text{FREQ:STAR}) \text{ @ } 28.75 \text{ kHz} - (\text{FREQ:SPAN}) / 2$$

FREQ:SPAN UP increases the frequency span to the next largest allowable value. FREQ:SPAN DOWN decreases the frequency span to the next smallest allowable value.

The frequency resolution is determined by the [SENSe:]FREQuency:RESolution command. The frequency span is proportional to the sampling rate. To increase the sampling rate, you must increase the frequency span.

$$\text{FREQ:SPAN} = \text{FREQ:RES} / \text{SWE:TIME}$$

In octave analysis instrument mode (INST:SEL OCT; Option 1D1):

This command is not valid. The frequency resolution is determined by the [SENSe:]FREQuency:RESolution:OCTave command.

In order analysis instrument mode (INST:SEL ORD; Option 1D0):

This command is not valid. The frequency resolution is determined by the frequency span.

$$\text{frequency resolution} = \text{FREQ:SPAN}/400$$

In swept sine instrument mode (INST:SEL SINE; Option 1D2):

The value specified by this command is rounded to the next higher 15.625 mHz step. The frequency span limits the range of the start frequency (FREQ:STAR) values according to the following formulas:

for two channel measurements:

$$(\text{FREQ:STAR}) \text{ @ } 51.2 \text{ kHz} - (\text{FREQ:SPAN})$$

for four channel measurements:

$$(\text{FREQ:STAR}) \text{ @ } 25.6 \text{ kHz} - (\text{FREQ:SPAN})$$

If logarithmic spacing is used (SWE:SPAC LOG), the value may be expressed in terms of hertz, decades or octaves.

In correlation analysis instrument mode (INST:SEL CORR):

This command is not valid.

[SENSe:]

Table 18-7. Range of Available Frequency Spans  
in FFT analysis instrument mode (in hertz)

Type of Analyzer	Type of Measurement			
	1 Channel (INPut2 OFF)		2 Channel (INPut2 ON;INPut 4 OFF)	4 Channel (INPut4 ON)
Standard 2 Channel	195 mHz - 102.4 kHz		97 mHz - 51.2 kHz	not available
Option AY6 4 Channel	AAF <sup>+</sup> off (INPut:FILTer OFF)	AAF <sup>+</sup> on ( INPut:FILTer ON)	97 mHz - 51.2 kHz	97 mHz - 25.6 kHz
	195 mHz - 102.4 kHz	195 mHz - 51.2 kHz		

+ AAF = anti alias filter



**[SENSe:]FREQUency:SPAN:FULL****command**

Sets the analyzer to the widest frequency span available for the current instrument mode.

Command Syntax: [SENSe:]FREQUency:SPAN:FULL

Example Statements: OUTPUT 711;"SENS:FREQUENCY:SPAN:FULL"  
OUTPUT 711;"freq:span:full"

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: confirmed

Description:

This command is not valid in order analysis or swept sine instrument modes.

In FFT analysis instrument mode (INST:SEL FFT) :

Number of Channels	Start Frequency	Stop Frequency
1 channel (without option AY6)	0 Hz	102.4 kHz
2 channel (without option AY6)	0 Hz	51.2 kHz
1 channel (with option AY6)	0 Hz	51.2 kHz
2 channel (with option AY6)	0 Hz	51.2 kHz
4 channel (with option AY6)	0 Hz	25.6 kHz

In Octave analysis instrument mode (INST:SEL OCT; Option 1D1) :

Number of Channels	Start Frequency	Stop Frequency
1 channel - 1/1	16 Hz	16 kHz
1 channel - 1/3	12.5 Hz	40 kHz
1 channel - 1/12	6.3825 Hz	12.338 kHz
2 channel - 1/1	16 Hz	8 kHz
2 channel - 1/3	12.5 Hz	20 kHz
2 channel - 1/12	6.3825 Hz	6.169 kHz
4 channel (Option AY6) - 1/1	16 Hz	4 kHz
4 channel (Option AY6) - 1/3	12.5 Hz	10 kHz
4 channel (Option AY6) - 1/12	6.3825 Hz	3.084 kHz

[SENSe:]

In Correlation analysis instrument mode (INST:SEL CORR) :

Number of Channels	Record Length
1 channel	7.8125 mS
2 channel	7.8125 mS
4 channel	15.625 mS

In Histogram/Time instrument mode (INST:SEL HIST) :

Number of Channels	Record Time
1 channel (without option AY6)	3.9062 mS
2 channel (without option AY6)	7.8125 mS
1 channel (with option AY6)	7.8125 mS
2 channel (with option AY6)	7.8125 mS
4 channel (with option AY6)	15.625 mS

---

**[SENSe:]FREQUency:SPAN:LINK****command/query**

Specifies the frequency parameter which remains constant if frequency span or record length is modified.

Command Syntax: [SENSe:] FREQUency:SPAN:LINK START | CENTer

Example Statements: OUTPUT 711; ":Freq:Span:Link CENTER"  
OUTPUT 711; "SENS:FREQUENCY:SPAN:LINK CENTER"

Query Syntax: [SENSe:] FREQUency:SPAN:LINK?

Return Format: STAR | CENT

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: START  
SCPI Compliance: confirmed

**Description:**

This command “anchors” or “fixes” the start frequency or the center frequency.

If FREQ:SPAN:LINK STAR is sent, the start frequency does not change when the frequency span or record length changes.

---

**Note**           FREQ:SPAN:LINK is set to CENTer after a reset (\*RST).

---

[SENSe:]

---

**[SENSe:]FREQUency:STARt**

**command/query**

Specifies the start (lowest) frequency for the frequency band of the current measurement.

Command Syntax: [SENSe:]FREQUency:STARt {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0.0:114999.9023  
<unit> ::= [HZ|CPM]  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"SENS:FREQ:STAR 1000"  
OUTPUT 711;":freq:star 4.8e3"

Query Syntax: [SENSe:]FREQUency:STARt?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +0.0  
SCPI Compliance: confirmed

Description:

In FFT analysis instrument mode (INST:SEL FFT) :

The values of FREQ:STAR and FREQ:SPAN define the frequency bandwidth. The size of the bandwidth (FREQ:SPAN) remains constant if the start frequency changes.

Allowable start frequency values are defined by the following formulas:

single channel measurements (without option AY6):

$$0 \leq \text{start frequency} \leq (115 \text{ kHz} - (\text{frequency span}/2))$$

two channel measurements or single channel measurements with option AY6:

$$0 \leq \text{start frequency} \leq (57.5 \text{ kHz} - (\text{frequency span}/2))$$

four channel measurements:

$$0 \leq \text{start frequency} \leq (28.75 \text{ kHz} - (\text{frequency span}/2))$$

Step size (FREQ:STEP) determines the change in frequency which results when you send UP or DOWN with this command.

See figure 18-2 for the range of start frequency values with different bandwidths.

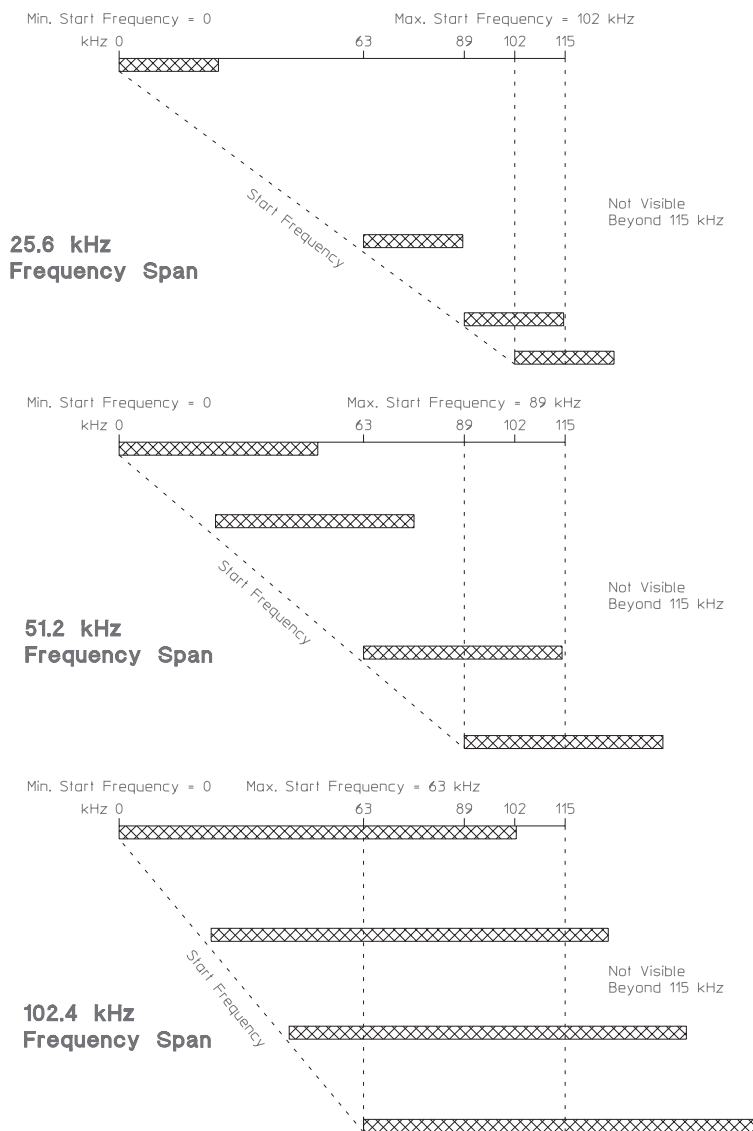


Figure 18-2. Allowable Start Frequency Values Depend On Bandwidth

In octave analysis instrument mode (INST:SEL OCT; Option 1D1):

This command specifies the start frequency for the octave measurement. The frequency can be specified in Hz or as a band number.

In swept sine instrument mode (INST:SEL SINE; Option 1D2):

The allowable values are: 15.625 mHz to 511199.984375 Hz. A value specified by this command is rounded to the next lower 15.625 mHz step.

[SENSe:]

---

**[SENSe:]FREQUency:STEP[:INCRement]**

**command/query**

Specifies the step size which is used when changing frequency parameters.

Command Syntax:            [SENSe:]FREQUency:STEP[:INCRement] {<number>[<unit>]} |<br>                            <step>|<bound>

<number> ::= a real number (NRf data)  
                              limits: 0.015625:102400.0

<unit> ::= [HZ|CPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;":FREQ:STEP 5e3"  
                            OUTPUT 711;"sens:freq:step:incr 100"

Query Syntax:             [SENSe:]FREQUency:STEP[:INCRement]?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                            Synchronization Required: no  
                            Preset State: +2.0E+003  
                            SCPI Compliance: confirmed

Description:

In FFT analysis instrument mode (INST:SEL FFT), step size determines the change in frequency which results when you send UP or DOWN with the FREQ:CENT or the FREQ:STAR commands.

In swept sine instrument mode (INST:SEL SINE; Option 1D2), step size determines the change in frequency which results when you send UP or DOWN with any of the following commands:

- FREQ:CENTer
- FREQ:STARt
- FREQ:STOP
- FREQ:MANual

**[SENSe:]FREQUency:STOP****command/query**

Sets the stop frequency to the specified value.

Command Syntax: [SENSe:]FREQUency:STOP {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: 0.03125:115000.0

<unit> ::= [HZ|CPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"SENS:FREQ:STOP 25.6 KHZ"  
OUTPUT 711;"frequency:stop max"

Query Syntax: [SENSe:]FREQUency:STOP?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +1.024E+005  
SCPI Compliance: confirmed

Description:

In FFT analysis instrument mode (INST:SEL FFT) :

This command defines the upper limit of the frequency bandwidth. The start frequency remains fixed (FREQ:SPAN:LINK). The values for the center frequency, the frequency span, and the record length change to appropriate values.

In octave measurement mode (INST:SEL OCT; Option 1D1) :

This command specifies the stop frequency for the octave measurement. The frequency can be specified in Hz or as a band number.

In swept sine instrument mode (INST:SEL SINE; Option 1D2) :

The value specified by this command is rounded to the next higher 15.625 mHz step. The start frequency (FREQ:STAR) is held constant and selected as the new "anchor" for the measurement. The center frequency and frequency span are adjusted to appropriate values.

[SENSe:]

---

## [SENSe:]HISTogram:BINs

command/query

Specifies the number of bins in a histogram.

Command Syntax: [SENSe:]HISTogram:BINs <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 4:1024  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;":HIST:BINs 10"  
                      OUTPUT 711;":SENSe:HISTOGRAM:BINs 400"

Query Syntax: [SENSe:]HISTogram:BINs?

Return Format: Real

Attribute Summary: Option: not applicable  
                      Synchronization Required: no  
                      Preset State: 512  
                      SCPI Compliance: instrument-specific

Description:

To obtain an optimal histogram, set the number of bins equal to the square root of the number of points specified with the [SENSe:]AVERage:TIME command.

$$\backslash \textit{Optimal Histogram} \approx ( \textit{HIST:BINs} ) = ( [ \textit{SENSe:} ] \textit{AVERage:TIME} ) \backslash$$



**[SENSe:]ORDer:MAXimum****command/query**

Specifies the number of orders to be displayed.

Command Syntax: [SENSe:]ORDer:MAXimum {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)

limits: 3.125:200

<unit> ::= [ORD]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"sens:ord:maximum 23.7537"  
OUTPUT 711;"Ord:Maximum 147.43"

Query Syntax: [SENSe:]ORDer:MAXimum?

Return Format: Real

Attribute Summary: Option: 1D0 Computed Order Tracking  
Synchronization Required: no  
Preset State: +10  
SCPI Compliance: instrument-specific

Description:

The command defines the highest order to be tracked and is used with the ORD:RESolution command to specify the spacing between order lines. The allowable range is 3.125 to 200 orders.

The displayed number of orders is limited as follows:

  
 ORD:MAX ® 200 (ORD:RES)

[SENSe:]

---

**[SENSe:]ORDER:RESolution**

**command/query**

Specifies order resolution.

Command Syntax:        [SENSe:]ORDER:RESolution {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: .0078125:1

<unit> ::= [ORD]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:    OUTPUT 711;" :order:resolution .5"  
                          OUTPUT 711;" SENS:ORD:RES 1"

Query Syntax:         [SENSe:]ORDER:RESolution?

Return Format:         Real

Attribute Summary:    Option: 1D0 Computed Order Tracking  
                          Synchronization Required: no  
                          Preset State: 0.1  
                          SCPI Compliance: instrument-specific

Description:

This command defines the spacing of the order map lines as a ratio of the number of orders displayed (ORD:MAX) divided by the number of lines per order.

$$\text{ORD:MAX} \leq 200 (\text{ORD:RES})$$

Values can range from .0078125 to 1.

**[SENSe:]ORDER:RESolution:TRACk****command/query**

Specifies the number of points per order track.

Command Syntax:            [SENSe:]ORDER:RESolution:TRACk <number>|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 1:2048
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements:        OUTPUT 711;":ord:resolution:trac 100"  
                           OUTPUT 711;"SENS:ORDER:RES:TRACK 50"

Query Syntax:             [SENSe:]ORDER:RESolution:TRACk?

Return Format:             Integer

Attribute Summary:        Option: 1D0 Computed Order Tracking  
                           Synchronization Required: no  
                           Preset State: +15  
                           SCPI Compliance: instrument-specific

Description:

This command specifies the resolution of your order track measurement.

See the [SENSe:]ORDER:TRACk command for information about specifying order track measurements.

[SENSe:]

---

**[SENSe:]ORDER:RPM:MAXimum**

**command/query**

Specifies the maximum rotational speed range you want to analyze.

Command Syntax:            [SENSe:] ORDER:RPM:MAXimum {<number>[<unit>]} |<step> |<bound>

<number> ::= a real number (NRf data)  
          limits: 0:9.9e37

<unit> ::= [RPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"ORD:RPM:MAX 5000"  
                              OUTPUT 711;"order:rpm:maximum 1e3"

Query Syntax:             [SENSe:] ORDER:RPM:MAXimum?

Return Format:            Real

Attribute Summary:        Option: 1D0 Computed Order Tracking  
                              Synchronization Required: no  
                              Preset State: +6000  
                              SCPI Compliance: instrument-specific

Description:

For runup measurements, the measurement stops at the speed specified by this command.

For rundown measurements, the measurement starts at the speed specified by this command and continues to the minimum RPM (ORD:RPM:MIN).

If the value you specify for ORD:RPM:MAX is less than the value set for ORD:RPM:MIN, the value for ORD:RPM:MIN is set to equal the value of ORD:RPM:MAX.

```
If
ORD:RPM:MAX < ORD:RPM:MIN
then
ORD:RPM:MIN = ORD:RPM:MAX
```

If the value you specify for ORD:RPM:MIN is greater than the value set for ORD:RPM:MAX, the value for ORD:RPM:MAX is set to equal the value of ORD:RPM:MIN.

```
If
ORD:RPM:MIN > ORD:RPM:MAX
then
ORD:RPM:MAX = ORD:RPM:MIN
```

**[SENSe:]ORDER:RPM:MINimum****command/query**

Specifies the minimum rotational speed range you want to analyze.

Command Syntax:           [SENSe:] ORDER:RPM:MINimum {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:9.9e37

<unit> ::= [RPM]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SENSE:ORDER:RPM:MINIMUM 800"  
                          OUTPUT 711;"ord:rpm:min 1E4"

Query Syntax:           [SENSe:] ORDER:RPM:MINimum?

Return Format:           Real

Attribute Summary:       Option: 1D0 Computed Order Tracking  
                          Synchronization Required: no  
                          Preset State: +600  
                          SCPI Compliance: instrument-specific

Description:

For runup measurements, the measurement starts at the speed specified by this command and continues to the maximum RPM (ORD:RPM:MAX).

For rundown measurements, the measurement stops at the speed specified by this command.

If the value you specify for ORD:RPM:MIN is greater than the value set for ORD:RPM:MAX, the value for ORD:RPM:MAX is set to equal the value of ORD:RPM:MIN.

```
If
ORD:RPM:MIN > ORD:RPM:MAX
then
ORD:RPM:MIN = ORD:RPM:MAX
```

If the value you specify for ORD:RPM:MAX is less than the value set for ORD:RPM:MIN, the value for ORD:RPM:MIN is set to equal the value of ORD:RPM:MAX.

```
If
ORD:RPM:MAX < ORD:RPM:MIN
then
ORD:RPM:MIN = ORD:RPM:MAX
```

[SENSe:]

---

**[SENSe:]ORDER:TRACk[1 | 2 | 3 | 4 | 5]**

**command/query**

Specify the order number for the selected track.

Command Syntax:            [SENSe:] ORDER:TRACk [1 | 2 | 3 | 4 | 5] {<number> [<unit>]} | <step> | <bound>

<number> := a real number (NRf data)  
          limits: 0:9.9e37

<unit> ::= [ORD]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"sens:order:track5 9"  
                              OUTPUT 711;"ORD:TRAC 3"

Query Syntax:             [SENSe:] ORDER:TRACk [1 | 2 | 3 | 4 | 5] ?

Return Format:            Real

Attribute Summary:        Option: 1D0 Computed Order Tracking  
                              Synchronization Required: no  
                              Preset State: +1 TRACK 1  
  +2 TRACK 2  
  +3 TRACK 3  
  +4 TRACK 4  
  +5 TRACK 5  
                              SCPI Compliance: instrument-specific

Description:

This command assigns order values to each of the five possible orders to be tracked. The value must be between 0 and the programmed highest order (ORD:MAX) and can be specified in .0001 increments.

If an order value is not assigned, the preset order number is used. See Preset State in the Attribute Summary above.

---

**[SENSe:]ORDer:TRACk[1 | 2 | 3 | 4 | 5]:STATe****command/query**

Selects order track or order spectrum measurements.

Command Syntax:           [SENSe:]ORDer:TRACk [1 | 2 | 3 | 4 | 5] :STATe OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;":ORD:TRAC2:STATE OFF"  
                              OUTPUT 711;"sens:order:trac2:stat OFF"

Query Syntax:             [SENSe:]ORDer:TRACk [1 | 2 | 3 | 4 | 5] :STATe?

Return Format:             Integer

Attribute Summary:        Option: 1D0 Computed Order Tracking  
                              Synchronization Required: no  
                              Preset State: +0 (all tracks)  
                              SCPI Compliance: instrument-specific

Description:

The analyzer makes an order track measurement when ORD:TRAC:STAT ON is sent.

The analyzer makes an order spectrum measurement when ORD:TRAC:STAT OFF is sent.

---

**Note**                    This command ignores the order track specifier. If the state is changed for one order track, the state changes for all order tracks.

---

[SENSe:]

---

**[SENSe:]REFErrence**

**command/query**

Specifies the reference channel(s).

Command Syntax:            [SENSe:]REFErrence SINGle|PAIR

Example Statements:        OUTPUT 711;"Sense:Ref PAIR"  
                              OUTPUT 711;"REFERENCE PAIR"

Query Syntax:              [SENSe:]REFErrence?

Return Format:              SING|PAIR

Attribute Summary:        Option:   AY6   Plus Two Channels  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

This command specifies the reference channel(s) for measurements using multiple channels.

Send REF SING to specify Channel 1 as the reference channel for channels 2, 3, and 4. For example, the CALC:FEED 'XFR:POW:RAT 3,1' command specifies a frequency response function with Channel 3 input data and Channel 1 as the reference.

Send REF PAIR to specify Channel 1 as the reference channel for Channel 2 and Channel 3 as the reference channel for Channel 4. With this selection, the only valid channel pairs are 2,1 and 4,3.

Use the INPut[1|2|3|4][:STATe] command to activate the input channels.

---

**Note**                      If REF PAIR is specified, Channel 1 is the only reference channel for Channel 2 and Channel 3 is the only reference channel for Channel 4. Channel 1 cannot be the reference channel for Channels 3 or 4.

---



**[SENSe:]REJect:STATe****command/query**

Turns overload rejection on or off.

Command Syntax: [SENSe:]REJect:STATe OFF|0|ON|1

Example Statements: OUTPUT 711;":rej:stat ON"  
OUTPUT 711;"Sense:Rej:State ON"

Query Syntax: [SENSe:]REJect:STATe?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF (+0)  
SCPI Compliance: instrument-specific

Description:

If overload rejection is off (the default condition), all time records are included in the measurement.

If SENS:REJ ON is sent, the time record from the overloaded input channel is not included in the measurement results. The concurrent time record from the other channel is rejected as well. The measurement continues until the analyzer has collected the specified number of non-overloaded time records.

[SENSe:]

---

**[SENSe:]SWEep:DIRection**

**command/query**

Specifies the direction of the sweep.

Command Syntax:            [SENSe:] SWEep:DIRection UP |DOWN

Example Statements:        OUTPUT 711;"SENS:SWE:DIRECTION UP"  
                              OUTPUT 711;"swe:direction UP"

Query Syntax:              [SENSe:] SWEep:DIRection?

Return Format:              UP |DOWN

Attribute Summary:        Option: 1D2 Swept Sine  
                              Synchronization Required: no  
                              Preset State: UP  
                              SCPI Compliance: confirmed

Description:

To initiate a sweep that begins at the lowest frequency (FREQ:START) and ends at the highest frequency (FREQ:STOP), send SWE:DIR UP.

To initiate a sweep that begins at the highest frequency (FREQ:STOP) and ends at the lowest frequency (FREQ:START), send SWE:DIR DOWN.

This command is not used if a manual sweep is specified with the SWEep:MODE MAN command.

**[SENSe:]SWEep:DWEL****command/query**

Specifies the integration time for swept sine measurements.

Command Syntax:            [SENSe:] SWEEP <dwel\_time>

                             <dwel\_time> ::= {<number>[<unit>]}|<step>|<bound>

                             <number> ::= a real number (NRf data)

    limits: 250e-6:32768 S

    or

    1:234 CYCLE

                             <unit> ::= [S|CYCLE]

                             <step> ::= UP|DOWN

                             <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"sense:sweep:dwel .005 s"

                             OUTPUT 711;"Swe:Dwell 1 cycle"

Query Syntax:             [SENSe:] SWEep:DWEL?

Return Format:             Real

Attribute Summary:        Option: 1D2 Swept Sine

                             Synchronization Required: no

                             Preset State: +5.0 CYCLE

                             SCPI Compliance: confirmed

Description:

Integration time is the amount of time that each point is measured.

Sending SWE:DWEL in seconds, results in a constant integration scale.

Sending SWE:DWEL in cycles, results in a proportional integration scale. At higher frequencies the same number of cycles occurs in a shorter time. The integrate time must be a minimum of 1 cycle long. The analyzer takes any value less than one as one complete cycle.

[SENSe:]

---

**[SENSe:]SWEep:MODE**

**command/query**

Specifies automatic or manual sweep modes.

Command Syntax: [SENSe:] SWEep:MODE AUTO|MANual

Example Statements: OUTPUT 711;":Swe:Mode AUTO"  
OUTPUT 711;"SENSE:SWE:MODE AUTO"

Query Syntax: [SENSe:] SWEep:MODE?

Return Format: AUTO|MAN

Attribute Summary: Option: 1D2 Swept Sine  
Synchronization Required: no  
Preset State: AUTO  
SCPI Compliance: confirmed

Description:

To select automatic sweep mode, send SWE:MODE AUTO. The instrument controls the sweep according to the following parameters:

- **FREQ:START**
- **FREQ:STOP**
- **FREQ:RESolution**
- **SWE:DIRection**
- **SWE:SPACe**

The values of the parameters are specified by default or by the appropriate command.

Send SWE:MODE MAN to select a discrete sweep; the measurement occurs only at the frequency points specified by the [SENSe:]FREQ:MANual command.

**[SENSe:]SWEep:OVERlap****command/query**

Specifies the maximum amount of time record overlap.

Command Syntax: [SENSe:]SWEep:OVERlap {<number>[<unit>]}|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 0:99
<unit>   ::= [PCT]
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements: OUTPUT 711;"sens:swe:overlap 52"  
OUTPUT 711;"Swe:Overlap 26"

Query Syntax: [SENSe:]SWEep:OVERlap?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +0 PCT  
SCPI Compliance: instrument-specific

Description:

Overlap processing is not used with triggered measurements. The analyzer must be in:

- real time
- automatic arming (ARM:SOUR IMM)
- frequency span ® 25.6 kHz (1 channel)  
frequency span ® 12.8 kHz channel  
frequency span ® 6.4 kHz (4 channel)  
(set with [SENSe:]FREQ:SPAN command)
- continuous (free run) trigger mode (TRIG:SOUR IMM)
- overload rejection off ([SENSe:]REJect:STATe OFF)

Data points from the end of one time record can be reused at the beginning of the next time record. This results in the overlapping of time records. Use this command to specify the amount of the block size which should be common to two consecutive time records.

[SENSe:]

As the frequency span decreases, the corresponding time record length increases. Overlap processing becomes possible when the instrument takes more time to collect time records than it does to process them. This allows you to make a faster measurement especially with narrow frequency spans. Overlap processing also reduces statistical variance caused by windowing.

You can specify overlap either as a percentage or as a fraction of the time record length. SWE:OVER 0.22 is the same as SWE:OVER 22 PCT. The value you send is rounded to the nearest allowable percentage (an integer between 0 and 99).

The query returns a value that indicates the amount of overlap currently specified. The value is returned as a percent.

**[SENSe:]SWEep:SPACing****command/query**

Selects linear or logarithmic spacing between measurement data points.

Command Syntax: [SENSe:]SWEep:SPACing LINear|LOGarithmic

Example Statements: OUTPUT 711;":SWE:SPAC LINEAR"  
OUTPUT 711;"sense:swe:spacing LOGARITHMIC"

Query Syntax: [SENSe:]SWEep:SPACing?

Return Format: LIN|LOG

Attribute Summary: Option: 1D2 Swept Sine  
Synchronization Required: no  
Preset State: LIN  
SCPI Compliance: confirmed

**Description:**

Send SWE:SPAC LIN for linearly spaced frequency points. The frequency step size does not change during the sweep of the frequency points; it remains constant over the entire spectrum.

Send SWE:SPAC LOG for logarithmically or proportionately spaced frequency points. The ratio of the location of adjacent points is constant. In addition to hertz, decade or octave units may be used with the FREQ:SPAN command. The following units may be used with the FREQ:RES command:

- PNT/SWP
- PNT/DEC
- PNT/OCT
- PCT (spacing between measurement points as a percentage of the total frequency span)

[SENSe:]

---

**[SENSe:]SWEep:STIME**

**command/query**

Specifies the settling time for a swept sine measurement.

Command Syntax: [SENSe:] SWEep:STIME {<number>[<unit>]} |<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:9.9e37  
<unit> ::= [S|CYCLE]  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"SENS:SWEEP:STIM 0.005 s"  
                      OUTPUT 711;"swe:stime 1 cycle"

Query Syntax: [SENSe:] SWEep:STIME?

Return Format: Real

Attribute Summary: Option: 1D2 Swept Sine  
                      Synchronization Required: no  
                      Preset State: +5.0 CYCLE  
                      SCPI Compliance: instrument-specific

Description:

Settling time is the delay between changing the source frequency and starting the measurement at each point. This allows the device under test to stabilize after the frequency changes.

Units can be specified in seconds (S) or cycles (CYCLE).



**[SENSe:]SWEep:TIME****command/query**

Specifies the length of the time record in seconds.

Command Syntax: [SENSe:] SWEep:TIME {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
 limits: 976.5625e-6:8192.0  
 <unit> ::= [S]  
 <step> ::= UP|DOWN  
 <bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"SWE:TIME 1 S"  
 OUTPUT 711;"sense:sweep:time 3.125e-2"

Query Syntax: [SENSe:] SWEep:TIME?

Return Format: Real

Attribute Summary: Option: not applicable  
 Synchronization Required: no  
 Preset State: +7.8049E-003  
 SCPI Compliance: confirmed

## Description:

When you send this command, two other values may be adjusted. The value of the frequency span (FREQ:SPAN) is adjusted to FREQ:RES / record length (in hertz). If the start frequency is “fixed” (FREQ:SPAN:LINK STAR), the center and stop frequencies are adjusted accordingly. If the center frequency is “fixed” (FREQ:SPAN:LINK CENT), the start and stop frequencies are adjusted.

If you change the frequency span, the time record length is adjusted to:

$$(FREQ:RES)/(FREQ:SPAN)$$

In correlation analysis instrument mode (INST:SEL CORR) :

This command specifies the record length, **T**. T represents the length of the raw time record collected. T is used in correlation windowing functions as follows:

-T/4 to T/4  
 0 to T/2  
 -T/2 to T/2

The record length, T, is independent of the frequency resolution (FREQ:RES). Changing the value of one does not change the value of the other. See the online help for additional information about record length in correlation analysis.

[SENSe:]

---

**[SENSe:]TCAPture:ABORt**

**command**

Stops the time capture process.

Command Syntax:            [SENSe:]TCAPture:ABORt

Example Statements:        OUTPUT 711;"Sens:Tcap:Abort"  
                              OUTPUT 711;"TCAP:ABORT"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The amount of data in the time capture buffer is less than the amount specified by the TCAP:LENG command. The analyzer aborts the time capture process immediately. Any partial time record is discard.

---

**[SENSe:]TCAPture:DElete****command**

Removes the time capture buffer.

Command Syntax:           [SENSe:]TCAPture:DElete

Example Statements:        OUTPUT 711;":tcap:del"  
                              OUTPUT 711;"Sense:Tcap>Delete"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

The analyzer removes the memory allocation for the time capture buffer, effectively “clearing” the buffer.

[SENSe:]

---

**[SENSe:]TCAPture:FILE**

**command/query**

Uploads or downloads time-capture file between the analyzer and the controller.

Command Syntax:            [SENSe:] TCAPture:FILE

                             <DATA> ::= <DEF\_BLOCK>

Example Statements:        OUTPUT 711;"SENSe:TCAP:FILE?"

Query Syntax:              [SENSe:] TCAPture:FILE?

Return Format:              DEF\_USER

Attribute Summary:        Option: not applicable  
                             Synchronization Required: no  
                             Preset state: not applicable  
                             SCPI Compliance: instrument-specific

Description:

This command is used to upload and download time-capture files between the analyzer and the controller. The main use of this command is for fast storage of time-capture files on a personal computer hard disc. See the XFER example program in appendix F.

The files sent to the controller are SDF (Standard Data Format) files. The files sent to the analyzer must also be SDF time-capture files.

Use the SENSe:DATA command to upload and download time-capture data.

**[SENSe:]TCAPture[:IMMediate]****command**

Starts the collection of data for the time capture process.

Command Syntax:            [SENSe:]TCAPture[:IMMediate]

Example Statements:        OUTPUT 711;"SENS:TCAP:IMMEDIATE"  
                              OUTPUT 711;"tcap"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The analyzer collects data from the input channels and stores it in the time capture buffer.

If the command (TCAP:MALL) which allocates memory has not been sent; the analyzer automatically allocates memory based on the current settings for the frequency span, the size of the time capture buffer (TCAP:LENG) and the size of the tachometer buffer if it is enabled with the [SENSe:]TCAP:TACH[:STATe] ON command.

The size of the tachometer buffer is determined by the number of revolutions per minute specified with the [SENSe:]TCAP:TACH:RPM:MAX command and the number of tachometer pulses per revolution specified with the TRIGger:TACH:PCOunt command.

[SENSe:]

---

**[SENSe:]TCAPture:LENGth**

**command/query**

Specifies the length of the time capture buffer.

Command Syntax: [SENSe:]TCAPture:LENGth {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:9.9e37  
<unit> ::= [S|BLK|PNT]  
<step> ::= UP|DOWN  
<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;" :Tcapture:Leng 4.54329e+36"  
OUTPUT 711;" SENS:TCAPTURE:LENG 2.91628e+37"

Query Syntax: [SENSe:]TCAPture:LENGth?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +0.0  
SCPI Compliance: instrument-specific

Description:

If length is specified in seconds, the size of the buffer is relative to the frequency span.

If length is specified in blocks or points, the size of the buffer is absolute. There are 1024 sample points per block.

Any specified length is rounded up to the nearest block.

The maximum capture length is dependent upon the memory configuration:

	Maximum Capture Length (in blocks of 1024 sample points)		
	1 Channel	2 Channel	4 Channel
Standard	900	400	200
Add 4 Mbytes RAM (Option AN2)	2900	1400	700
Add 8 Mbytes RAM (Option UFC)	5000	2500	1200

The minimum is 1 block. If the TCAP:ABOR command is sent, the analyzer aborts the process and discards the partially filled block.

---

**Note**            The analyzer's memory is not allocated until TCAP:MALL or TCAP[:IMM] has been sent.

---

[SENSe:]

---

**[SENSe:]TCAPture:MALLocate**

**command**

Allocates memory for the time capture buffer.

Command Syntax:            [SENSe:]TCAPture:MALLocate

Example Statements:        OUTPUT 711;"sense:tcap:mall"  
                              OUTPUT 711;"Tcapture:Mall"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The analyzer automatically allocates memory when the TCAP[:IMM] command is sent. So although this command is not necessary, it ensures you have sufficient memory allocated for the time capture buffer.



**[SENSe:]TCAPture:STARt[1 | 2 | 3 | 4]****command/query**

Specifies the beginning of the time capture data used in a measurement.

Command Syntax: [SENSe:]TCAPture:STARt [1 | 2 | 3 | 4] {<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: -9.9e37:9.9e37

<unit> ::= [S|BLK|PNT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"Sens:Tcap:Star 1 mS"  
OUTPUT 711;"TCAPTURE:START 3 BLK"

Query Syntax: [SENSe:]TCAPture:STARt [1 | 2 | 3 | 4] ?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +0.0  
SCPI Compliance: instrument-specific

Description:

This command allows to select a portion of the time capture data to be used for the measurement. You can set the “analysis region” for each channel independently.

Use the TCAP:STOP command to specify the end of the time capture data.

The analyzer sets this value to the beginning of the capture data upon receiving any of the following commands:

- MMEMory:LOAD:STATe
- MMEMory:LOAD:TCAPture
- [SENSe:]TCAPture[:IMMEDIATE]
- SYSTem:PRESet
- \*RST

[SENSe:]

---

**[SENSe:]TCAPture:STOP[1 | 2 | 3 | 4]**

**command/query**

Specifies the end of the time capture data used in a measurement.

Command Syntax:            [SENSe:] TCAPture:STOP [1 | 2 | 3 | 4] {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:9.9e37

<unit> ::= [S|BLK|PNT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;":tcap:stop3 10 blk"  
                          OUTPUT 711;"sense:tcapture:stop 4 s"

Query Syntax:             [SENSe:] TCAPture:STOP [1 | 2 | 3 | 4] ?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: end of data  
                          SCPI Compliance: instrument-specific

Description:

This command allows you to select a portion of the time capture data to be used for the measurement. You can set the “analysis region” for each channel independently.

Use the SENS:TCAP:START command to specify the beginning of the time capture data.

The analyzer sets this value to the end of the capture data upon receiving any of the following commands:

- MMEMory:LOAD:STATe
- MMEMory:LOAD:TCAPture
- [SENSe:]TCAPture[:IMMEDIATE]
- SYSTem:PRESet
- \*RST

**[SENSe:]TCAPture:TACHometer:RPM:MAXimum****command/query**

Specifies the tachometer's maximum RPM when included in the time capture buffer. \$Itachometer;maximum RPM in time capture

Command Syntax: [SENSe:]TCAPture:TACHometer:RPM:MAXimum <number>|<step>|<bound>

```
<number> ::= a real number (NRf data)
           limits: 5:491519
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements: OUTPUT 711;" :TCAPTURE:TACH:RPM:MAXIMUM 193783"  
OUTPUT 711;"sens:tcapture:tach:rpm:maximum 275254"

Query Syntax: [SENSe:]TCAPture:TACHometer:RPM:MAXimum?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +6000  
SCPI Compliance: instrument-specific

**Description:**

This command sets the upper limit of the rotation speed range you want to monitor for measurements using the time capture buffer.

In a runup measurement, this value specifies when the measurement stops. In a rundown measurement, this value specifies when the measurement starts.

---

**Note** This command is not used when the analyzer is in order analysis (INST:SEL ORD; Option 1D0). The analyzer's tachometer is always ON in this instrument mode. Specify the tachometer's maximum RPM with the [SENSe:]ORDER:RPM:MAXimum command when the analyzer is in order analysis instrument mode.

---

To include the tachometer input signal in the time capture buffer, send the TCAPture:TACHometer[:STATe] command.

---

**Caution** The value specified with this command is used by the analyzer to allocate memory for the tachometer buffer. If the value specified is too low, the analyzer aborts the time capture when it fills the tachometer buffer.

---

[SENSe:]

---

**[SENSe:]TCAPture:TACHometer[:STATe]**

**command/query**

Directs the analyzer to include the tachometer input signal in the time capture buffer.

Command Syntax: [SENSe:]TCAPture:TACHometer[:STATe] OFF|0|ON|1

Example Statements: OUTPUT 711;"Sens:Tcapture:Tach:Stat OFF"  
OUTPUT 711;"TCAPTURE:TACH ON"

Query Syntax: [SENSe:]TCAPture:TACHometer[:STATe]?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: OFF (+0)  
SCPI Compliance: instrument-specific

Description:

This command determines if the tachometer input signal is included in the time capture buffer.

The tachometer parameters must be setup before capturing the data. See the TCAP:TACH:RPM:MAX and TRIGger:TACHometer commands for more information.

---

**Note** This command is not required for order analysis (INST:SEL ORD; Option 1D0). The analyzer's tachometer is always ON in order analysis instrument mode.

---

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:AUTO****command/query**

Automatically selects the best range on the specified channel for the current input signal.

Command Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:AUTO OFF | 0 | ON | 1

Example Statements: OUTPUT 711;"sens:volt3:dc:range:auto off"  
OUTPUT 711;"VOLTAGE:RANGE:AUTO 1"

Query Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:AUTO?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: ON (+1)  
SCPI Compliance: confirmed

**Description:**

This command automatically adjusts the input range. Two types of autorange are available; up/down (VOLT:RANG:AUTO:DIR EITHER) and up only (VOLT:RANG:AUTO:DIR UP). The default selection is up/down.

If up/down autorange is selected, the analyzer searches for the best input range at the start of a measurement. If the analyzer detects an overload condition, the analyzer steps the input up through successive range values (+2 dB increments) until the input is no longer in an overload condition. If the analyzer detects that the input signal has dropped below half the range value, the analyzer steps the input down in -2 dB increments until the input has returned to half range.

If up only autorange is selected, the analyzer selects the lowest input range at the start of a measurement. If the analyzer detects an overload condition, the analyzer steps the input up through successive range values (+2 dB increments) until the input is no longer in an overload condition. The analyzer never adjusts the range downward in response to a decrease in signal amplitude. If the range is too large for the current input signal, send VOLT[1|2|3|4]:RANG:AUTO ON to restart autorange.

If you use this command for a swept sine measurement, the analyzer adjusts the input range upward or downward at each measurement point, depending on the signal level at the measurement point.

---

**Note** The analyzer does not autorange during a time capture or during an averaged measurement.

---

[SENSe:]

To turn off the autorange feature:

- Set the input range by specifying a value with the VOLT:RANG[:UPPer] command.

OR

- Send VOLT[1|2|3|4]:RANG:AUTO OFF. The range is fixed at the last autorange value.

If the channel specifier is not used, the command defaults to channel 1.

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:AUTO:DIRection****command/query**

Selects the type of autorange; up/down or up only.

Command Syntax:	[SENSe:]VOLTage [1   2   3   4] [:DC] :RANGe:AUTO:DIRection UP EITHer
Example Statements:	OUTPUT 711;"SENSE:VOLTAGE3:DC:RANGE:AUTO:DIRECTION UP" OUTPUT 711;"volt:rang:;auto:dir eith"
Query Syntax:	[SENSe:]VOLTage [1   2   3   4] [:DC] :RANGe:AUTO:DIRection?
Return Format:	UP EITH
Attribute Summary:	Option: not applicable Synchronization Required: no Preset State: EITHer SCPI Compliance: instrument-specific

**Description:**

This command specifies the type of autorange used by the analyzer when VOLTage:DC:RANGe:AUTO is ON.

To set the analyzer so it adjusts the input range upward or downward, send VOLT:DC:RANG:AUTO:DIR EITHer. If the analyzer detects an overload condition it will adjust the input range up. If the analyzer detects an input signal that has dropped below half the range, it will adjust the input range down.

To set the analyzer so it adjusts the input range upward and only upward send VOLT:DC:RANG:AUTO:DIR UP. The analyzer never adjusts the range downward in response to a decrease in signal amplitude. If the range is too large for the current input signal you must send the VOLTage:DC:RANGe:AUTO ON command to restart the autorange.

In swept sine instrument mode (INST SINE), the analyzer ignores this command. The analyzer adjusts the input range upward or downward at each measurement point, depending on the signal level at the measurement point.

[SENSe:]

---

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UNIT:USER:LABel**

**command/query**

Assigns a name to the transducer units for the specified input channel when VOLT:RANG:UNIT:XDCR:LABel is USER.

Command Syntax: [SENSe:]VOLTage:RANGe:UNIT:USER:LABel '<STRING>'

<STRING> ::= ASCII characters - 32 through 126  
maximum number of characters: 4

Example Statements: OUTPUT 711;"SENSe:VOLT2:RANG:UNIT:USER:LABel 'watt'"  
OUTPUT 711;"volt:rang:unit:user:lab 'Cdeg'"

Query Syntax: [SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UNIT:USER:LABel?

Return Format: STRING

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: EU  
SCPI Compliance: instrument-specific

Description:

The label appears only when the VOLT:RANG:UNIT:XDCR command is sent with the USER parameter.

If the channel specifier is not used, the command defaults to channel 1.

The query returns the last-entered transducer unit name for the specified channel.

---

**Note** Only use this command if the label you want is not one of the VOLT:RANG:UNIT:XDCR:LABel parameters.

---



**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UNIT:USER:SFACTOR****command/query**

Specifies the transducer sensitivity for transducer units.

Command Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:USER:SFACTOR  
{<number> [<unit>] } | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: -9.9e37:9.9e37

<unit> ::= [V/EU|EU/V]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"volt:rang:unit:user:sfac 0.1"  
OUTPUT 711:"SENS:VOLTAGE3:DC:RANGE:UNIT:USER:SFACTOR  
1.8e-06"

Query Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:USER:SFACTOR?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

## Description:

This command sets the sensitivity of a mechanical-to-electrical transducer as the ratio between the electrical signal (output) and mechanical quantity (input). The ratio is Volts per transducer unit.

The transducer sensitivity and the transducer unit label are used only when VOLT:RANG:UNIT:USER:STATe is ON.

If the channel specifier is not used, the command defaults to channel 1.

Refer to the documentation for your transducer for the appropriate sensitivity value.

[SENSe:]

---

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UNIT:USER[:STATe]**

**command/query**

Enables the use of transducer units.

Command Syntax:            [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:USER [:STATe]  
                              OFF | 0 | ON | 1

Example Statements:        OUTPUT 711;":voltage:rang:unit:user OFF"  
                              OUTPUT 711;"Sens:Voltage:Dc:Rang:Unit:User:State OFF"

Query Syntax:              [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:USER [:STATe] ?

Return Format:             Integer

Attribute Summary:        Option:    not applicable  
                              Synchronization Required:    no  
                              Preset State:    OFF (+0)  
                              SCPI Compliance:    instrument-specific

Description:

This command allows you to specify a transducer unit for each input channel.

When VOLT:RANG:UNIT:USER is OFF, the unit is Volts.

When VOLT:RANG:UNIT:USER is ON, the transducer unit is specified by the VOLT:RANG:UNIT:XDCR command and the transducer sensitivity is specified by the VOLT:RANG:UNIT:USER:SFACTor command. If the channel specifier is not used, the command defaults to channel 1.

To use transducer units:

1. Specify a unit label.

- a. To select a label for sound pressure, acceleration, velocity, or displacement, use the VOLT:RANG:UNIT:XDCR:LABel command. Parameters include Pascals, g's, meters, meters per second, meters per second<sup>2</sup>, kilograms, Newtons, dynes, inches, inches per second, inches per second<sup>2</sup>, pounds, and mils.

- b. To define your own label, set VOLT:RANG:UNIT:XDCR:LABel to USER and send the VOLT:RANG:UNIT:LABel command with your own label.

2. Specify the transducer's sensitivity with the VOLT:RANG:UNIT:USER:SFACTor command.

3. Enable the transducer unit setup with VOLT:RANG:UNIT:USER ON.

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UNIT:XDCR:LABel** **command/query**

Specifies a transducer unit for the specified channel.

Command Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:XDCR:LABel  
PA | G | M/S2 | M/S | M | KG | N | DYN | INCH/S2 | INCH/S | INCH | MIL | LB | USER

Example Statements: OUTPUT 711;"SENSe:VOLT2:RANG:UNIT:XDCR:LABel PA"  
OUTPUT 711;"volt:rang:unit:user:lab G"

Query Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe:UNIT:XDCR:LABel?

Return Format: PA | G | M/S2 | M/S | M | KG | N | DYN | INCH/S2 | INCH/S | INCH | LB | MIL | USER

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: USER  
SCPI Compliance: instrument-specific

## Description:

Select a unit label appropriate for your transducer. The transducer unit label appears only when VOLT:RANG:UNIT:USER is ON.

## Examples of Transducer Units

	Unit of Measure	Command
microphones	Pascals	VOLT:RANG:UNIT:XDCR PA
accelerometers	g's	VOLT:RANG:UNIT:XDCR G
displacement probes	mils	VOLT:RANG:XDCR MIL
impact hammers	pound force	VOLT:RANG:XDCR LB
shakers	Newtons	VOLT:RANG:XDCR N

This command works with the CALC:UNIT:MECH command. If transducer units are G, M/S2, M/S, M, INCH/S2, INCH/S, INCH, or MIL; the analyzer automatically integrates or differentiates the input data using the CALC:UNIT:MECH command. If the transducer unit is PA, the analyzer automatically displays dB SPL as the Y-axis unit for dB magnitude coordinate transforms.

If your transducer measures in units other than pressure, acceleration, velocity, or displacement; you will have to set this command to USER (VOLT:RANG:UNIT:XDCR:LAB USER) and define your own label with the VOLT:RANG:UNIT:USER:LABel command.

If the channel specifier is not used, the command defaults to channel 1. The query returns the last-entered transducer unit name for the specified channel.

[SENSe:]

---

**[SENSe:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe[:UPPer]**

**command/query**

Specifies the input range for the selected channel.

Command Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe [:UPPer] {<number> [<unit>] } |<step> |<bound>

<number> ::= a real number (NRf data)  
limits: -51:31.66

<unit> ::= [DBVRMS | VPK | DBVPK | V | DBV | EU | DBEU | VRMS]

<step> ::= UP | DOWN

<bound> ::= MAX | MIN

Example Statements: OUTPUT 711;":volt4:rang -51"  
OUTPUT 711;":SENS:VOLT1:DC:RANG:UPP DOWN"

Query Syntax: [SENSe:]VOLTage [1 | 2 | 3 | 4] [:DC] :RANGe [:UPPer] ?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: -51 dBVrms  
SCPI Compliance: confirmed

Description:

This command sets the range for the input channel. Valid input ranges are from 27 through -51 dBVrms in 2 dB steps. If you send a value that is not allowed, it is rounded up to the next higher value. If you do not send specify units when you send a new value, the default unit is used, DBVRMS.

If the channel specifier is not used, the command defaults to channel 1.

To increment the value of the input range to the next higher value (+2 dB), send VOLT:RANG UP. To decrement the value of the input range to the next lower value (-2 dB), send VOLT:RANG DOWN. To set the input range to a value near to the amplitude of the main marker value, send VOLT[1|2|3|4][:DC]:RANG (CALC[1|2]:MARK:Y?).

To determine units send the query, VOLT[1|2|3|4]:RANG? UNIT. See the [SENSe:]VOLTage[:DC]:RANGe:AUTO command for information about the analyzer's autorange feature.

---

**Note** You can specify a new value for a channel range regardless of the analyzer's current channel setting (1, 2 or 4 channels). For example, you can specify a new value for the channel 2 range while you are in one channel instrument mode. However, the value is not used to set the channel range until you enter the appropriate channel mode.

---

**[SENSe:]WINDow[1 | 2 | 3 | 4]:EXPonential****command/query**

Specifies the time constant for the exponential window function.

Command Syntax: [SENSe:]WINDow[1|2|3|4]:EXPonential {<number>[<unit>]} |  
<step> | <bound>

<number> ::= a real number (NRf data)  
limits: 3.8147E-6:9.9999E6

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"Sense:Window:Exponential .1"  
OUTPUT 711;"WIND:EXP 2.5"

Query Syntax: [SENSe:]WINDow[1|2|3|4]:EXPonential?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +9.999E+003  
SCPI Compliance: confirmed

Description:

The time constant is used to calculate the exponential decay for the exponential window according to the following formula:

$$e^{-t/\tau}$$

Where

t is the position (in time) in the record length

$\tau$  is the time constant

The channel specifier, WINDow[1|2|3|4], is ignored. When the force window is specified (WIND FORC), the exponential time constant is applied to all channels—even when the exponential window is not specified.

---

**Note** The first point in the time record is always considered to be time t=0.

---

[SENSe:]

---

**[SENSe:]WINDow[1 | 2 | 3 | 4]:FORCe**

**command/query**

Specifies the width of the force window.

Command Syntax:            [SENSe:]WINDow [1 | 2 | 3 | 4] :FORCe {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: 3.8147E-6:9.9999E6

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;" :WINDOW:FORCE 0.1 s"  
                              OUTPUT 711;"SENS:WIND:FORC 2 ms"

Query Syntax:             [SENSe:]WINDow [1 | 2 | 3 | 4] :FORCe?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +9.999E+003 S  
                              SCPI Compliance: confirmed

Description:

This command specifies the length of the force window in seconds.

The force window passes the first part of the time record (specified by the length of the width of the force window) and sets the remaining part to the average value of the time record's remaining data. The data is then multiplied by the exponential windowing function.

The channel specifier, WINDow[1|2|3|4], is ignored. The force width is applied to all channels.

---

**Note**                    The first point in the time record is always considered to be time t=0.

---

**[SENSe:]WINDow[1 | 2 | 3 | 4]:ORDer:DC****command/query**

Directs the analyzer to include the DC bin in the composite power calculation (order track measurements).

Command Syntax: [SENSe:]WINDow[1|2|3|4]:ORDer:DC OFF|0|ON|1

Example Statements: OUTPUT 711;":WIND:ORD:DC OFF  
OUTPUT 711;"window:order:dc 1"

Query Syntax: [SENSe:]WINDow[1|2|3|4]:ORDer:DC?

Return Format: Integer

Attribute Summary: Option: 1D0 Computed Order Tracking  
Synchronization Required: no  
Preset State: ON (+1)  
SCPI Compliance: instrument-specific

## Description:

The composite power calculation (CALC:FEED 'XFR:POW:COMP') sums the power of the order spectrum at each RPM step.

To exclude the power of the order spectrum in the DC bin, send WIND:ORD:DC OFF.

The first bin is excluded from the composite power calculation if the measurement is using a uniform windowing function (WINDow:TYPE UNIF).

The first two bins are excluded from the calculation if the measurement is using a Hann window (WINDow:TYPE HANN).

The first five bins are excluded from the calculation if the measurement is using a flattop window (WINDow:TYPE FLAT).

---

**Note** This command is not channel specific. It ignores the channel specifier.

---

[SENSe:]

---

**[SENSe:]WINDow[1 | 2 | 3 | 4][:TYPE]**

**command/query**

Selects the type of windowing function.

Command Syntax: [SENSe:]WINDow [1 | 2 | 3 | 4] [:TYPE] HANNing | FLATtop | UNIFORM | FORCe | EXPonential | LAG | LLAG

Example Statements: OUTPUT 711;"SENS:WIND:TYPE HANN  
OUTPUT 711;"WINDOW FLAT

Query Syntax: [SENSe:]WINDow [1 | 2 | 3 | 4] [:TYPE] ?

Return Format: HANN | FLAT | UNIF | FORC | EXP | LAG | LLAG

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: FLAT  
SCPI Compliance: confirmed

Description:

To select a Hanning window, send WIND HANN. The beginning and end of the time record have a zero value which forces a periodic form on the data. It is commonly used to measure random noise and provides better frequency resolution.

To select a flattop window, send WIND FLAT. This window function is similar to the a Hanning window, but is optimized for narrow band signals with a flatter passband. It has increased amplitude accuracy but less frequency resolution.

To select a uniform window, send WIND UNIF. The entire time record is weighted uniformly—effectively, a windowing function is not applied. This window function should be used for signals which may be considered self-windowing, such as transients, bursts and periodic waveforms.

The channel specifier is ignored for the Hanning, flattop, and uniform window functions.

To select a force window, send WIND FORC. This window function is a modified uniform window. It passes the input signal for the specified amount of time (WINDow[1|2|3|4]:FORCe) then attenuates it to the average value of the remaining data for the remainder of the time record. Typically, the force window is only applied to the reference channel(s).

To select the exponential window, send WIND EXP. This function attenuates the input signal at a decaying exponential rate determined by the specified time constant (WINDow[1|2|3|4]:EXPonential).

The force window is always multiplied by the exponential window— even when the exponential window is not selected.

You can specify the force or exponential window function for each channel with the channel specifier, WINDow[1|2|3|4]. The default combination of force/exponential windowing (after Preset) is force window for channel 1 and exponential window for channels 2, 3, and 4.



In order analysis instrument mode (INST:SEL ORD; Option 1D0) you may specify one of the following windowing functions:

- Hanning
- flattop
- uniform

The windowing function is not available in octave analysis instrument mode (INST:SEL OCT; Option 1D1) and swept sine instrument mode (INST:SEL SINE; Option 1D2).

In correlation analysis instrument mode (SEL:INST CORR), this command specifies the correlation weighting function.

To select the uniform function, send WIND UNIF. The uniform weighting function (- T/2, T/2) does not suppress any part of the time record. This function should be used for signals which may be considered self-windowing, such as transients, bursts and periodic waveforms.

To select the Zero Pad 0, T/2 function, send WIND LAG. The function suppresses the last half of the time record and passes only the first half.

To select the Zero Pad -T/4, T/4 function, send WIND LLAG. The function suppresses the first quarter and the last quarter of the time record, and passes the center part of the time record (the second and third quarters).

The channel specifier is ignored for the uniform, Zero Pad 0,T/2 (LAG), and Zero Pad -T/4,T4 (LLAG) window functions.

See Online Help for additional information about correlation's weighting functions.



SOURce

---



## SOURce

---

Commands in this subsystem control the analyzer's source output. See the OUTPut subsystem for commands which enable the analyzer's source output.

**SOURce:BURSt****command/query**

Sets the burst length for the burst source types.

Command Syntax:           SOURce:BURSt {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:100

<unit> ::= [PCT]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"SOUR:BURS 26.2401"  
                              OUTPUT 711;"source:burs 91.7044"

Query Syntax:             SOURce:BURSt?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +50 PCT  
                              SCPI Compliance: instrument-specific

Description:

This command is used with the SOUR:FUNC BRAN and SOUR:FUNC BCH commands.

The active time of the burst cycle, "burst length," is set as a percentage of the total time record. The burst starts at the beginning of the time record.

The query returns a value in percent.

**SOURce:FREQuency[:CW]****command/query**

Sets the frequency of the sine source.

Command Syntax:           SOURce:FREQuency[:CW] {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)

          limits: 0.0:115000.0

<unit> ::= [HZ]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"sour:freq 1 khz"  
                          OUTPUT 711;"SOURCE:FREQUENCY:CW 12000"

Query Syntax:             SOURce:FREQuency[:CW] ?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +1.024E+004  
                          SCPI Compliance: confirmed

Description:

This command sets the frequency of the fixed sine source type. To select a fixed sine source output, use the SOUR:FUNC SIN command.

The allowable range is 0 to 115 kHz. The frequency may be set to 15.625 mHz increments.

This is an alias for the SCPI command SOURce:FREQuency:FIXed.

**SOURce:FREQuency:FIXed****command/query**

Sets the frequency of the sine source type.

Command Syntax:           SOURce:FREQuency:FIXed {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0.0:115000.0  
<unit>    ::= [HZ]  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;"sour:freq:fix 1.128 khz"  
                          OUTPUT 711;"SOURCE:FREQUENCY:FIXED 5000"

Query Syntax:             SOURce:FREQuency:FIXed?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +1.024E+004  
                          SCPI Compliance: confirmed

Description:

This command sets the frequency of the fixed sine source type. To select a fixed sine source output, use the SOUR:FUNC SIN command.

The allowable range is 0 to 115 kHz. The frequency may be set to 15.625 mHz increments.

This is an alias for the SCPI command SOURce:FREQuency[:CW].



**SOURce:FUNCtion[:SHAPE]****command/query**

Specifies the source output.

Command Syntax:           SOURce:FUNCtion[:SHAPE] SINusoid|RANDom|BRANdom|PCHirp|  
BCHirp|PINK|USER|CAPT

Example Statements:        OUTPUT 711;":Source:Func CAPT"  
OUTPUT 711;":SOUR:FUNCTION:SHAP BRANDOM"

Query Syntax:             SOURce:FUNCtion[:SHAPE]?

Return Format:            SIN|RAND|BRAN|PCH|BCH|PINK|USER|CAPT

Attribute Summary:        Option:   not applicable  
Synchronization Required: no  
Preset State:    SIN  
SCPI Compliance:  confirmed

## Description:

To select a sinusoidal waveform, send SOUR:FUNC SIN. Refer to the SOURce:FREQuency commands for information about setting the frequency for the sine waveform.

To select random noise, send SOUR:FUNC RAND. Random noise is a continuous gaussian distributed noise signal. The signal is band-limited and band-translated to concentrate the energy in the frequency span defined by the commands in the [SENse:]FREQ subsystem.

To select burst random noise, send SOUR:FUNC BRAN. Burst random noise is a gaussian distributed noise signal in successive bursts. The SOUR:BURS command is used to specify the burst length as a percentage of the time record. This is an instrument-specific command.

To select periodic chirp, send SOUR:FUNC PCH. Periodic chirp is a fast sine sweep across the current frequency span. The sweep repeats with the same period as the current time record.

To select burst chirp, send SOUR:FUNC BCH. Burst chirp is a fast sine sweep over the current frequency span for a portion of the time record. The SOUR:BURS command is used to specify the burst length as a percentage of the time record. This is an instrument-specific command.

To select pink noise, send SOUR:FUNC PINK. Pink noise is noise whose spectral density is inversely proportional to frequency. This is an instrument-specific command.

To select arbitrary source data, send SOUR:FUNC USER. Option 1D4, Arbitrary Source, must be installed. The data must be stored in a data register specified with the SOURce:USER[:REGister] command.

To select time capture data, send SOUR:FUNC CAPT. Option 1D4, Arbitrary Source, must be installed. Use the SOURce:USER:CAPTure command to indicate which channel of time capture data to use. This is an instrument-specific command.

**SOURce:USER:CAPTure****command/query**

Specifies which channel of time capture data to use.

Command Syntax:           SOURce:USER:CAPTure <number>|<step>|<bound>

    <number> ::= a real number (NRf data)

                  limits: 1:4

    <step> ::= UP|DOWN

    <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"source:user:capt 3"  
                          OUTPUT 711;"Source:User:Capture 1"

Query Syntax:             SOURce:USER:CAPTure?

Return Format:            Integer

Attribute Summary:        Option: 1D4 Arbitrary Source  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

The parameter specifies which input channel of time capture data the analyzer is to use for the SOURce:FUNCTION:SHAPE CAPTure command.

**SOURce:USER[:REGister]****command/query**

Specifies the data register which contains the data for the arbitrary source.

Command Syntax:           SOURce:USER[:REGister] D1|D2|D3|D4|D5|D6|D7|D8

Example Statements:        OUTPUT 711;":SOUR:USER D3"  
                          OUTPUT 711;"source:user:register D7"

Query Syntax:             SOURce:USER[:REGister]?

Return Format:             D1|D2|D3|D4|D5|D6|D7|D8

Attribute Summary:        Option:   1D4 Arbitrary Source  
                          Synchronization Required: no  
                          Preset State: D1  
                          SCPI Compliance: instrument-specific

**Description:**

This command is used with the SOUR:FUNC USER command which sets the source to output arbitrary data. Send SOUR:USER to identify the data register location of the arbitrary waveform.

Refer to the MMEMory:LOAD:TRACe, and TRACe[:DATA] commands for additional information about saving arbitrary source data.

---

**SOURce:USER:REPeat****command/query**

Enables the source repeat function.

Command Syntax:           SOURce:USER:REPeat OFF|0|ON|1

Example Statements:        OUTPUT 711;"Sour:User:Repeat OFF"  
                              OUTPUT 711;"SOUR:USER:REP ON"

Query Syntax:             SOURce:USER:REPeat?

Return Format:             Integer

Attribute Summary:        Option: 1D4 Arbitrary Source  
                              Synchronization Required: no  
                              Preset State: ON (+1)  
                              SCPI Compliance: instrument-specific

Description:

To output the arbitrary source data only during data collection of the measurement, send SOURCE:REPEAT OFF. One “waveform” is present in each measurement record.

To output the arbitrary source data continuously, send SOURCE:USER:REPEAT ON. The source “repeats” the arbitrary waveform. This is the selection at Preset.

---

**Note**                    This command has no effect when the analyzer is in free run trigger mode (TRIG:SOUR IMM).

---

**SOURce:VOLTage[:LEVel]:AUTO****command/query**

Enables the autolevel feature in swept sine measurements (INST:SEL SINE).\$!source;autolevel

Command Syntax:           SOURce:VOLTage[:LEVel]:AUTO OFF|0|ON|1

Example Statements:        OUTPUT 711;":sour:voltage:auto OFF"  
                               OUTPUT 711;"Source:Volt:Lev:Auto ON"

Query Syntax:             SOURce:VOLTage[:LEVel]:AUTO?

Return Format:             Integer

Attribute Summary:        Option: 1D2 Swept Sine  
                               Synchronization Required: no  
                               Preset State: OFF (+0)  
                               SCPI Compliance: instrument-specific

## Description:

If autolevel is enabled with the SOUR:VOLT:AUTO ON command, the analyzer adjusts the source output level to keep the amplitude of an input channel within a specified range.

See the following commands for additional information about setting the range:

- SOURce:VOLTage:LIMit[:AMPLitude]
- SOURce:VOLTage:LIMit:INPut
- SOURce:VOLTage[:LEVel]:REFerence
- SOURce:VOLTage[:LEVel]:REFerence:CHANnel
- SOURce:VOLTage[:LEVel]:REFerence:TOLerance

To turn off the autolevel function, send SOUR:VOLT:AUTO OFF.

Refer to Online Help for detailed information about the autolevel feature.

**SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]****command/query**

Specifies the source output level.

Command Syntax:           SOURce:VOLTage[:LEVel] [:IMMediate] [:AMPLitude] {<number> [ <unit> ] } | <step> | <bound>

<number> ::= a real number (NRf data)  
          limits: -9.9e37:13.9794

<unit> ::= [DBVRMS|VPK|DBVPK|V|DBV|VRMS]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SOUR:VOLT 0 DBV"  
                          OUTPUT 711;"sour:volt:lev:imm:amp 5 VPK"

Query Syntax:            SOURce:VOLTage[:LEVel] [:IMMediate] [:AMPLitude] ?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.0 VPK  
                          SCPI Compliance: confirmed

Description:

The source output level can be expressed in terms of peak values, Vpk, or rms values, Vrms.

While most source waveforms are limited to a maximum of 5 VPK, pink noise (SOUR:FUNC PINK) is limited to 4.196 VPK. The minimum is 0 VPK. If the source output level is specified in DBVRMS, the smallest non-zero level is -74.912 DBVRMS.

To determine if the output level is set to PEAK, PP (peak-to-peak), or RMS values, send SOUR:VOLT? UNIT. Each source waveform has a specific peak-to-rms value. The maximum rms level depends upon the source waveform. See Online Help for additional information.

**SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet****command/query**

Specifies a DC offset for the source output.

Command Syntax:           SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet {<number>  
                                  [<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -10.0:10.0

<unit> ::= [V]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"SOUR:VOLTAGE:LEV:IMM:OFFSET 9.40924"  
                                  OUTPUT 711;"sour:voltage:offs -2.92258"

Query Syntax:             SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet?

Return Format:            Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +0.0 V  
                              SCPI Compliance: confirmed

Description:

Use this command to set a DC bias on the analyzer's source.

The allowable range is -10 V to + 10 V. If the DC offset is between -2 V and +2 V, it can be set in 1 mV increments; otherwise it is set in 5 mV increments.

The analyzer resets the DC offset level to the nearest valid increment if the the sum of the source level (set with the SOURce:VOLTage[:LEVel][:IMMediate][:AMPlitude] command) and the DC offset exceed  $\pm 10$  Volts.

**SOURce:VOLTage[:LEVel]:REFerence****command/query**

Specifies the amplitude of the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE).

Command Syntax:           SOURce:VOLTage[:LEVel]:REFerence {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: -69.276:31.66  
          <unit> ::= [DBVRMS|VPK|DBVPK|V|DBV|EU|DBEU|VRMS]  
          <step> ::= UP|DOWN  
          <bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SOUR:VOLT:LEV:REF:CHAN .1 VPK"  
                          OUTPUT 711;"sour:volt:ref:chan 2 VRMS"

Query Syntax:           SOURce:VOLTage[:LEVel]:REFerence?

Return Format:           Real

Attribute Summary:       Option: 1D2 Swept Sine  
                          Synchronization Required: no  
                          Preset State: +2.0 VPK  
                          SCPI Compliance: instrument-specific

## Description:

This command specifies the amplitude which the analyzer tries to maintain for the input reference channel when autolevel is enabled (SOUR:VOLT:AUTO ON).



**SOURce:VOLTage[:LEVel]:REFerence:CHANnel****command/query**

Selects the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE).

Command Syntax:           SOURce:VOLTage[:LEVel]:REFerence:CHANnel INPut1|INPut2|  
                              INPut3|INPut4

Example Statements:        OUTPUT 711;":Sour:Voltage:Ref:Channel INPUT2"  
                              OUTPUT 711;"SOUR:VOLT:LEVEL:REF:CHANNEL INPUT1"

Query Syntax:             SOURce:VOLTage[:LEVel]:REFerence:CHANnel?

Return Format:             INP1|INP2|INP3|INP4

Attribute Summary:        Option: 1D2 Swept Sine  
                              Synchronization Required: no  
                              Preset State: INP2  
                              SCPI Compliance: instrument-specific

**Description:**

This command specifies which input channel the analyzer monitors when the autolevel is enabled (SOUR:VOLT:AUTO ON). The analyzer adjusts the source output to keep the amplitude of this input channel within the range specified with the SOURce:VOLTage[:LEVel]:REFerence.

**SOURce:VOLTage[:LEVel]:REFerence:TOLerance****command/query**

Specifies the sensitivity of the autolevel algorithm in swept sine measurements (INST:SEL SINE).

Command Syntax:           SOURce:VOLTage[:LEVel]:REFerence:TOLerance {<number>  
                                  [<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0.1:20

<unit> ::= [DB]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"sour:volt:level:ref:tolerance 4.38485"  
                                  OUTPUT 711;"Sour:Volt:Reference:Tol 6.80982"

Query Syntax:             SOURce:VOLTage[:LEVel]:REFerence:TOLerance?

Return Format:            Real

Attribute Summary:       Option: 1D2 Swept Sine  
                                  Synchronization Required: no  
                                  Preset State: +2.0  
                                  SCPI Compliance: instrument-specific

## Description:

This command sets a tolerance band (in dB) for the autolevel algorithm.

If the amplitude of the reference input channel falls outside of the specified range (relative to the value set with the SOUR:VOLT[:LEV]:REF command), the analyzer adjusts the amplitude of the source output when SOUR:VOLT:AUTO ON.

**SOURce:VOLTage:LIMit[:AMPLitude]****command/query**

Sets the maximum limit used by the autolevel algorithm to adjust the source's amplitude in swept sine measurements (INST:SEL SINE).

Command Syntax:           SOURce:VOLTage:LIMit[:AMPLitude] {<number>  
                                  [<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -9.9e37:13.9794  
  
<unit> ::= [DBVRMS|VPK|DBVPK|V|DBV|VRMS]  
  
<step> ::= UP|DOWN  
  
<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"Source:Voltage:Limit 0 DBV"  
                                  OUTPUT 711;"sour:volt:lim 1 VRMS"

Query Syntax:             SOURce:VOLTage:LIMit[:AMPLitude]?

Return Format:            Real

Attribute Summary:        Option: 1D2 Swept Sine  
                                  Synchronization Required: no  
                                  Preset State: +2.0 VPK  
                                  SCPI Compliance: confirmed

## Description:

This command controls the analyzer's autolevel algorithm in adjusting the source's output level during a sweep. The limit constrains any autolevel adjustment to the source's amplitude.

**SOURce:VOLTage:LIMit:INPut****command/query**

Sets the maximum amplitude of the non-reference input channels for the autolevel feature in swept sine measurements (INST:SEL SINE).

Command Syntax:         SOURce:VOLTage:LIMit:INPut {<number>[<unit>]}|  
                              <step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: -69.276:31.66  
<unit>    ::= [DBVRMS|VPK|DBVPK|V|DBV|EU|DBEU|VRMS]  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;" :SOURCE:VOLT:LIM:INPUT -23.4139"  
                              OUTPUT 711;"sour:voltage:lim:inp -64.7616"

Query Syntax:             SOURce:VOLTage:LIMit:INPut?

Return Format:            Real

Attribute Summary:        Option: 1D2 Swept Sine  
                              Synchronization Required: no  
                              Preset State: +2.0 VPK  
                              SCPI Compliance: instrument-specific

**Description:**

This command controls the amplitude of the “other” input channels during a sweep. (The input channels not selected as the reference channel with the SOURce:VOLTage[:LEVel]:REFerence:CHANnel command).

The limit constrains any adjustment to the source’s output attempted by autolevel.

**SOURce:VOLTage:SLEW****command/query**

Specifies the source amplitude ramp rate in swept sine measurements (INST:SEL SINE).

Command Syntax:           SOURce:VOLTage:SLEW {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)

          limits: 0.0:10000.0

<unit> ::= [V/S|VPK/S|VRMS/S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SOUR:VOLT:SLEW 0 VRMS/S"  
                          OUTPUT 711;"source:voltage:slew .1 VPK/S"

Query Syntax:            SOURce:VOLTage:SLEW?

Return Format:           Real

Attribute Summary:       Option: 1D2 Swept Sine  
                          Synchronization Required: no  
                          Preset State: +0.0 VPK/S  
                          SCPI Compliance: confirmed

Description:

This command allows you to specify how fast the source amplitude changes when you start or stop a swept sine measurement.



STATus

---





# STATus

---

Commands in this subsystem provide access to most of the Agilent 35670A’s status groups (register sets). Some of the common commands described in chapter 3 provide access to the other register sets.

Most of the commands in this subsystem are used to set bits in status registers. Most of the queries are used to read status registers. Decimal weights are assigned to bits according to the following formula:

$$\text{weight} = 2^n$$

where n is the bit number with acceptable values of 0 through 14.

To set a single register bit to 1, send the decimal weight of that bit with the command that writes the register. See figure 20-1. To set more than one bit to 1, send the sum of the decimal weights of all the bits. Queries that read registers always return the sum of the decimal weights of all bits that are currently set to 1.

Bit Weights

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
X	

Figure 20-1. Table of Bit Weights

See “Using Status Registers” in chapter 1 for more information on the Agilent 35670A status groups.

---

**Note**            The STATus commands are listed alphabetically. Therefore, the STATus:QUESTionable:NTR command and the STATus:QUESTionable:PTR command follow the STATus:QUESTionable:LIMit commands.

---

---

**STATus:DEvice:CONDition?****query**

Reads and clears the Device State condition register.

Query Syntax:                STATus:DEvice:CONDition?

Example Statements:        OUTPUT 711;":status:dev:condition?"  
                              OUTPUT 711;"Stat:Dev:Condition?"

Return Format:               Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Device State condition register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See “Device State Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPIB Programmer’s Guide* for information about the role of condition registers in register sets.

**STATus:DEVIce:ENABle****command/query**

Sets and queries bits in the Device State enable register.

Command Syntax:           STATus:DEVIce:ENABle <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"stat:dev:enab 4"  
                          OUTPUT 711;":STATUS:DEVICE ENABLE 1"

Query Syntax:             STATus:DEVIce:ENABle?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

Description:

To set a single bit in the Device State enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is *not* modified when you send the \*RST command.

See "Device State Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the GPIB Programmer's Guide for information about the role of enable registers in register sets.

---

**STATus:DEvice[:EVENT]?****query**

Reads and clears the Device State event register.

Query Syntax:                 STATus:DEvice[:EVENT]?

Example Statements:         OUTPUT 711;"STAT:DEVICE:EVEN?"  
                               OUTPUT 711;"stat:device?"

Return Format:                Integer

Attribute Summary:         Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: instrument-specific

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Device State event register. (The decimal weight of a bit is  $2^n$ , where  $n$  is the bit number.)

---

**Note**                    The Device State event register is automatically cleared after it is read by this query.

---

See “Device State Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPIB Programmer’s Guide* for information about the role of event registers in register sets.

**STATus:DEVIce:NTRansition****command/query**

Sets and queries bits in the Device Status negative transition register.

Command Syntax:           STATus:DEVIce:NTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":stat:dev:ntr 4"  
                          OUTPUT 711;"STATUS:DEVICE:NTR 17"

Query Syntax:             STATus:DEVIce:NTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

## Description:

To set a single bit in the Device Status negative transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is *not* modified when you send the \*RST command.

See "Device State Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of negative transition registers in register sets.

**STATus:DEVIce:PTRansition****command/query**

Sets and queries bits in the Device State positive transition register.

Command Syntax:           STATus:DEVIce:PTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"status:dev:ptransition 17"  
                          OUTPUT 711;"stat:dev:ptr 20"

Query Syntax:             STATus:DEVIce:PTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

## Description:

To set a single bit in the Device State positive transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Device State Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of positive transition registers in register sets.

---

**STATus:OPERation:CONDition?****query**

Reads the Operation Status condition register.

Query Syntax:                 STATus:OPERation:CONDition?

Example Statements:         OUTPUT 711;":Stat:Operation:Cond?"  
                               OUTPUT 711;"STAT:OPERATION:COND?"

Return Format:                Integer

Attribute Summary:         Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Operation Status condition register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See "Operation Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of condition registers in register sets.

**STATus:OPERation:ENABLE****command/query**

Sets and queries bits in the Operation Status enable register.

Command Syntax:           STATus:OPERation:ENABLE <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"Status:Oper:Enab 96"  
                              OUTPUT 711;"status:operation:enable 2"

Query Syntax:             STATus:OPERation:ENABLE?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

Description:

To set a single bit in the Operation Status enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Operation Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of enable registers in register sets.



**STATus:OPERation[:EVENT]?****query**

Reads and clears the Operation Status event register.

Query Syntax:               STATus:OPERation[:EVENT]?

Example Statements:        OUTPUT 711;"status:oper:even?"  
                              OUTPUT 711;"Status:Oper?"

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Operation Status event register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

---

**Note**                    The Operation Status event register is automatically cleared after it is read by his query.

---

See “Operation Status Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPiB Programmer’s Guide* for information about the role of event registers in register sets.

**STATus:OPERation:NTRansition****command/query**

Sets and queries bits in the Operation Status negative transition register.

Command Syntax:           STATus:OPERation:NTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"STAT:OPER:NTR 260"  
                          OUTPUT 711;"status:operation:ntransition 16386"

Query Syntax:             STATus:OPERation:NTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

To set a single bit in the Operation Status negative transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Operation Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of negative transition registers in register sets.

**STATus:OPERation:PTRansition****command/query**

Sets bits in the Operation Status positive transition register.

Command Syntax:           STATus:OPERation:PTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"status:operation:ptransition 1536"  
                              OUTPUT 711;"STAT:OPER:PTR 2048"

Query Syntax:             STATus:OPERation:PTRansition?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

## Description:

To set a single bit in the Operation Status positive transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Operation Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of positive transition registers in register sets.

**STATus:PRESet****command**

Sets bits in most enable and transition registers to their default state.

Command Syntax:           STATus:PRESet

Example Statements:        OUTPUT 711;":STATUS:PRES"  
                              OUTPUT 711;"stat:preset"

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

STAT:PRES has the following effect on the Limit Fail and Questionable Voltage register sets:

- Sets all enable register bits to 1.
- Sets all positive transition register bits to 1.
- Sets all negative transition register bits to 0.

STATUS:PRESet has the effect of bringing all events to the second level register sets (the Device State, Questionable Status, and Operation Status) without creating an SRQ or reflecting events in a serial poll.

It also affects these register sets, (the Device State, Questionable Status, and Operation Status) as follows:

- Sets all enable register bits to 0.
- Sets all positive transition register bits to 1.
- Sets all negative transition register bits to 0.

STAT:PRES sets all bits in the User Defined enable register to 0. It does not affect any other register.

---

**STATus:QUEStionable:CONDition?****query**

Reads and clears the Questionable Status condition register.

Query Syntax:                   STATus:QUEStionable:CONDition?

Example Statements:            OUTPUT 711;"Stat:Questionable:Cond?"  
                                  OUTPUT 711;"STAT:QUESTIONABLE:COND?"

Return Format:                 Integer

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Questionable Status condition register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See "Questionable Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of condition registers in register sets.

**STATus:QUEStionable:ENABle****command/query**

Sets and queries bits in the Questionable Status enable register.

Command Syntax:           STATus:QUEStionable:ENABle <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"stat:ques:enab 1"  
                          OUTPUT 711;"stat:questionable:enable 513"

Query Syntax:             STATus:QUEStionable:ENABle?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

To set a single bit in the Questionable Status enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of enable registers in register sets.

---

**STATus:QUEStionable[:EVENT]?****query**

Reads and clears the Questionable Status event register.

Query Syntax:               STATus:QUEStionable[:EVENT]?

Example Statements:        OUTPUT 711;":status:ques?"  
                              OUTPUT 711;"Stat:Questionable:Even?"

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Questionable Status event register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

---

**Note**                    The Questionable Status event register is automatically cleared after it is read by this query.

---

See “Questionable Status Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPiB Programmer’s Guide* for information about the role of event registers in register sets.

---

**STATus:QUEStionable:LIMit:CONDition?**

**query**

Reads and clears the Limit Fail condition register.

Query Syntax:                    STATus:QUEStionable:LIMit:CONDition?

Example Statements:            OUTPUT 711;"STATUS:QUES:LIM:CONDITION?"  
                                   OUTPUT 711;"stat:questionable:lim:cond?"

Return Format:                   Integer

Attribute Summary:            Option:   not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Limit Fail condition register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See “Limit Fail Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPIB Programmer’s Guide* for information about the role of condition registers in register sets.



**STATus:QUEStionable:LIMit:ENABle****command/query**

Sets and queries bits in the Limit Fail enable register.

Command Syntax:           STATus:QUEStionable:LIMit:ENABle <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"stat:ques:lim:enab 3"  
                          OUTPUT 711;"STATUS:QUES:LIMIT:ENABLE 15"

Query Syntax:             STATus:QUEStionable:LIMit:ENABle?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

Description:

To set a single bit in the Limit Fail enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

All bits are initialized to 0 when the analyzer is turned on. However, the current setting of bits is not modified when you send the \*RST command.

See "Limit Fail Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIO Programmer's Guide* for information about the role of enable registers in register sets.

---

**STATus:QUEStionable:LIMit[:EVENT]?**

**query**

Reads and clears the Limit Fail event register.

Query Syntax:                    STATus:QUEStionable:LIMit[:EVENT]?

Example Statements:            OUTPUT 711;":Status:Ques:Limit?"  
                                   OUTPUT 711;"STAT:QUES:LIMIT:EVENT?"

Return Format:                   Integer

Attribute Summary:            Option:   not applicable  
                                   Synchronization Required: no  
                                   Preset State: not applicable  
                                   SCPI Compliance: instrument-specific

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Limit Fail event register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

---

**Note**                    The Limit Fail event register is automatically cleared after it is read by this query.

---

See “Limit Fail Register Set” in chapter 1 for a definition of bits of bits in the register set. See “Programming the Status System” in the *GPiB Programmer’s Guide* for information about the role of event registers in register sets.

**STATus:QUEStionable:LIMit:NTRansition****command/query**

Sets and queries bits in the Limit Fail negative transition register.

Command Syntax:           STATus:QUEStionable:LIMit:NTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :STAT:QUESTIONABLE:LIMIT:NTRANSITION 2"  
                          OUTPUT 711;"status:ques:lim:ntr 7"

Query Syntax:             STATus:QUEStionable:LIMit:NTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

## Description:

To set a single bit in the Limit Fail negative transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Limit Fail Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of negative transition registers in register sets.

**STATus:QUEStionable:LIMit:PTRansition****command/query**

Sets queries bits in the Limit Fail positive transition register.

Command Syntax:           STATus:QUEStionable:LIMit:PTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Status:Ques:Limit:Ptransition 4"  
                          OUTPUT 711;"stat:ques:lim:ptr 12"

Query Syntax:             STATus:QUEStionable:LIMit:PTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

## Description:

To set a single bit in the Limit Fail positive transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Limit Fail Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the  *GPIB Programmer's Guide*  for information about the role of positive transition registers in register sets.

**STATus:QUEStionable:NTRansition****command/query**

Sets and queries bits in the Questionable Status negative transition register.

Command Syntax:           STATus:QUEStionable:NTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"status:ques:ntnr 1"  
                          OUTPUT 711;"Status:Questionable:Ntransition 256"

Query Syntax:             STATus:QUEStionable:NTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

## Description:

To set a single bit in the Questionable Status negative transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of negative transition registers in register sets.

**STATus:QUEStionable:PTRansition****command/query**

Sets and queries bits in the Questionable Status positive transition register.

Command Syntax:           STATus:QUEStionable:PTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"STAT:QUESTIONABLE:PTR 256"  
                          OUTPUT 711;"status:ques:ptr 768"

Query Syntax:             STATus:QUEStionable:PTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

To set a single bit in the Questionable Status positive transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Status Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of positive transition registers in register sets.

---

**STATus:QUEStionable:VOLTagE:CONDition?****query**

Reads the Questionable Voltage condition register.

Query Syntax:                STATus:QUEStionable:VOLTagE:CONDition?

Example Statements:        OUTPUT 711;"status:ques:volt:condition?"  
                              OUTPUT 711;"Stat:Questionable:Volt:Cond?"

Return Format:               Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Questionable Voltage condition register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See “Questionable Voltage Register Set” in chapter 1 for a definition of bits in the register set. See “Programming the Status System” in the *GPiB Programmer’s Guide* for information about the role of condition registers in register sets.

**STATus:QUEStionable:VOLTage:ENABLE****command/query**

Sets and queries bits in the Questionable Voltage enable register.

Command Syntax:           STATus:QUEStionable:VOLTage:ENABLE <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"status:ques:voltage:enable 1"  
                          OUTPUT 711;"STAT:QUES:VOLT:ENAB 3"

Query Syntax:             STATus:QUEStionable:VOLTage:ENABLE?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

## Description:

To set a single bit in the Questionable Voltage enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Voltage Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of enable registers in register sets.



---

**STATus:QUEStionable:VOLTage[:EVENT]?****query**

Reads and clears the Questionable Voltage event register.

Query Syntax:                STATus:QUEStionable:VOLTage[:EVENT]?

Example Statements:        OUTPUT 711;":STATUS:QUES:VOLTAGE?"  
                              OUTPUT 711;"stat:ques:voltage:even?"

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the Questionable Voltage event register. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

---

**Note**                    The Questionable Voltage event register is automatically cleared after it is read by this query.

---

See "Questionable Voltage Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of event registers in register sets.

**STATus:QUEStionable:VOLTage:NTRansition****command/query**

Sets and queries bits in the Questionable Voltage negative transition register.

Command Syntax:           STATus:QUEStionable:VOLTage:NTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"stat:ques:volt:ntnr 2"  
                              OUTPUT 711;"STATUS:QUESTIONABLE:VOLTAGE:NTRANSITION 1"

Query Syntax:             STATus:QUEStionable:VOLTage:NTRansition?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

## Description:

To set a single bit in the Questionable Voltage negative transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Voltage Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of negative transition registers in register sets.

**STATus:QUEStionable:VOLTage:PTRansition****command/query**

Sets bits in the Questionable Voltage positive transition register.

Command Syntax:           STATus:QUEStionable:VOLTage:PTRansition <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":STAT:QUES:VOLTAGE:PTR 2"  
                          OUTPUT 711;"status:ques:volt:ptr 3"

Query Syntax:             STATus:QUEStionable:VOLTage:PTRansition?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

## Description:

To set a single bit in the Questionable Voltage positive transition register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "Questionable Voltage Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of positive transition registers in register sets.

**STATus:USER:ENABLE****command/query**

Sets and queries bits in the User Defined enable register.

Command Syntax:           STATus:USER:ENABLE <number>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Status:User:Enab 18066"  
                          OUTPUT 711;"STATUS:USER:ENABLE 10359"

Query Syntax:             STATus:USER:ENABLE?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: instrument-specific

Description:

To set a single bit in the User Defined enable register to 1, send the bit's decimal weight with this command. To set more than one bit to 1, send the sum of the decimal weights of all the bits. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

All bits are initialized to 0 when the analyzer is turned on or when the STAT:PRES command is sent. However, the current setting of bits is not modified when you send the \*RST command.

See "User Defined Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPIB Programmer's Guide* for information about the role of enable registers in register sets.

---

**STATus:USER[:EVENT]?****query**

Reads and clears the User Defined event register.

Query Syntax:               STATus:USER[:EVENT]?

Example Statements:        OUTPUT 711;":stat:user?"  
                              OUTPUT 711;"Status:User:Event?"

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

This query returns the sum of the decimal weights of all bits currently set to 1 in the User Defined event register. (The decimal weight of a bit is  $2^n$ , where  $n$  is the bit number.)

---

**Note**                    The User Defined event register is automatically cleared after it is read by this query.

---

See "User Defined Register Set" in chapter 1 for a definition of bits in the register set. See "Programming the Status System" in the *GPiB Programmer's Guide* for information about the role of event registers in register sets.

**STATus:USER:PULSe****command**

Sets bits in the User Defined event register.

Command Syntax:           STATus:USER:PULSe <number>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:32767

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"status:user:pulse 17664"  
                          OUTPUT 711;":STAT:USER:PUL 1856"

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: instrument-specific

Description:

Each bit in the User Defined event register is set to 1 when you send the bit's decimal weight with the STAT:USER:PULS command. (The decimal weight of a bit is  $2^n$ , where n is the bit number.)

See the "User Defined Register Set" in chapter 1 for more information.

SYSTem

---





## SYSTem

---

Commands in this subsystem are not related to analyzer performance. Instead, the SYSTem commands control global functions such as instrument preset, time and date.

**SYSTem:BEEPer[:IMMEDIATE]**

**command**

Generates the tone at a given frequency from the analyzer's beeper.

Command Syntax:           SYSTem:BEEPer[:IMMEDIATE] [<frequency>[,<time>[,<volume>]]]

```

<frequency> ::= [<number>|<bound>]
  <number> ::= a real number (NRf data)
               limits: 25:14000
  <bound> ::= MAX|MIN
  <time> ::= [<number>|<bound>]
  <number> ::= a real number (NRf data)
               limits: 0:60
  <bound> ::= MAX|MIN
  <volume> ::= [<number>|<bound>]
  <number> ::= a real number (NRf data)
               limits: 0.0:1.0
  <bound> ::= MAX|MIN
    
```

Example Statements:       OUTPUT 711;"SYST:BEEP 200"  
                           OUTPUT 711;":system:beep:immediate 4000, 5"

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

Description:

The frequency is specified in Hertz. The duration is specified in seconds. The volume parameter is accepted but is not used.

The frequency defaults to 2500 Hz and the time defaults to 0.23 seconds.

**SYSTem:BEEPer:STATe****command/query**

Enables the analyzer's beeper.

Command Syntax:           SYSTem:BEEPer:STATe OFF|0|ON|1

Example Statements:        OUTPUT 711;":syst:beep:state ON"  
                              OUTPUT 711;"Syst:Beeper:Stat OFF"

Query Syntax:             SYSTem:BEEPer:STATe?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  ON (+1)  
                              SCPI Compliance:  confirmed

**Description:**

When the beeper is enabled, it emits an audible tone when some messages are either displayed or placed in the error queue. It also emits an audible tone when a trace falls outside its specified limits if limit testing and the limit-fail beeper is turned on (CALCulate:LIMit:STATe ON and CALCulate:LIMit:BEEP ON).

When the beeper is disabled, the SYST:BEEP command will not generate any sound.

---

**SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS****command/query**

Sets the analyzer's GPIB address.

Command Syntax:           SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0:30

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;" :SYST:COMMUNICATE:GPIB:SELF:ADDRESS 0"  
                          OUTPUT 711;"syst:comm:gpiB:addr 3"

Query Syntax:           SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS?

Return Format:           Integer

Attribute Summary:      Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

**Description:**

The analyzer's address is saved in non-volatile memory, so it is retained when you turn the analyzer off and on.

---

**Note**                   When you use this command, wait at least 5 seconds before sending another command to the new address.

---

**SYSTem:COMMunicate:SERial[:RECeive]:BAUD****command/query**

Specifies the data-transfer rate between the analyzer and RS-232-C peripheral devices.serial;;baud rate

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:BAUD <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 300:9600

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;":SYST:COMM:SER:BAUD 9600"  
                          OUTPUT 711;"system:comm:serial:receive:baud 300"

Query Syntax:            SYSTem:COMMunicate:SERial[:RECeive]:BAUD?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

## Description:

This command specifies the baud rate between the analyzer and any RS-232-C peripherals.

Legal values are 300, 1200, 2400, 4800, and 9600.

**SYSTem:COMMunicate:SERial[:RECeive]:BITS****command/query**

Specifies the number of bits in a character for the RS-232-C interface.

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:BITS <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 5:8

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SYST:COMM:SER:REC:BITS 7"  
                          OUTPUT 711;"SYST:COMM:SER:BITS 8"

Query Syntax:            SYSTem:COMMunicate:SERial[:RECeive]:BITS?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

Legal values are 5, 6, 7 or 8.

---

**SYSTem:COMMunicate:SERial[:RECeive]:PACE****command/query**

Sets the RS-232-C receiver handshake pacing type.

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:PACE NONE|XON

Example Statements:        OUTPUT 711;"SYST:COMM:SER:REC:PACE XON"  
                          OUTPUT 711;"system:comm:serial:pace none"

Query Syntax:             SYSTem:COMMunicate:SERial[:RECeive]:PACE?

Return Format:            NONE|XON

Attribute Summary:        Option:   not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

**Description:**

This command specifies the type of handshake the analyzer uses when it is receiving data.

XON specifies the XON/XOFF protocol. The analyzer sends XOFF (DC3, decimal code 19) to the peripheral to stop the transmission of data. When it is able to receive additional data, the analyzer sends XON (DC1, decimal code 17) to the peripheral, which then resumes transmitting data. This software handshake is the easiest to implement.

NONE specifies the absence of a handshake protocol for pacing.

Check the documentation for your peripheral device to verify it supports the specified protocol.

**SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECK**

**command/query**

Enables RS-232-C parity verification capability.

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECK  
OFF|0|ON|1

Example Statements:        OUTPUT 711;"system:communicate:serial:parity:check ON"  
OUTPUT 711;"SYST:COMM:SER:REC:PAR:CHEC 1"

Query Syntax:             SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECK?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
Synchronization Required: no  
Preset State:   not affected by Preset  
SCPI Compliance:  confirmed

Description:

This command turns on parity verification for the RS-232-C interface. To set the type of parity, send the SYSTem:COMMunicate:SERial:PARity command.



**SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]****command/query**

Sets the parity generated for characters transmitted over the RS-232-C interface.

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]  
NONE | EVEN | ODD

Example Statements:        OUTPUT 711;"SYST:COMM:SER:PAR ODD"  
OUTPUT 711;"syst:communicate:serial:receive:parity none"

Query Syntax:             SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]?

Return Format:            NONE | EVEN | ODD

Attribute Summary:        Option:   not applicable  
Synchronization Required: no  
Preset State:   not affected by Preset  
SCPI Compliance:  confirmed

## Description:

Send SYST:COMM:SERIAL:PARITY ODD, if you want ODD parity.

Send SYST:COMM:SERIAL:PARITY EVEN, if you want EVEN parity.

Send SYST:COMM:SERIAL:PARITY NONE, if a parity bit is *not* to be included.

To turn on parity verification, send the SYSTem:COMMunicate:SERial:RECeive:PARity:CHECK ON command.

**SYSTem:COMMunicate:SERial[:RECeive]:SBITs****command/query**

Specifies the number of “stop bits” sent with each character over the RS-232-C interface.

Command Syntax:           SYSTem:COMMunicate:SERial[:RECeive]:SBITs <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 1:2

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"SYST:COMM:SER:REC:SBIT 1"  
                          OUTPUT 711;"system:communicate:serial:sbits 2"

Query Syntax:            SYSTem:COMMunicate:SERial[:RECeive]:SBITs?

Return Format:            Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not affected by Preset  
                          SCPI Compliance: confirmed

Description:

Legal values are 1 and 2.

**SYSTem:COMMunicate:SERial:TRANsmit:PACE****command/query**

Sets the RS-232 transmitter handshake pacing type.

Command Syntax:           SYSTem:COMMunicate:SERial:TRANsmit:PACE NONE | XON | DSR

Example Statements:        OUTPUT 711;"SYST:COMM:SER:TRAN:PACE XON"  
                               OUTPUT 711;"system:comm:serial:transmit:pace none"

Query Syntax:             SYSTem:COMMunicate:SERial:TRANsmit:PACE?

Return Format:             NONE | XON | DSR

Attribute Summary:        Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not affected by Preset  
                               SCPI Compliance: confirmed, except DSR

Description:

This command specifies the handshake type when the analyzer is sending data.

XON specifies the XON/XOFF protocol. The peripheral device must send XOFF (DC3, decimal code 19) to the analyzer to stop the transmission of data. When the peripheral device is ready to accept more data, it must send XON (DC1, decimal code 17) to the analyzer, which then resumes transmission. This software handshake is the easiest to implement.

DSR specifies a hardwire handshake. When the peripheral device is able to receive data, it sets its Data Terminal Ready line true (high). This sets the analyzer's Data Set Ready (DSR, pin 6) line true. When the peripheral device is not able to receive data, it sets its DTR line false (low). The analyzer will not send data until the peripheral device's DTR line is set true.

NONE specifies the absence of a handshake protocol for pacing.

Check the documentation for your peripheral device to verify it supports the specified protocol.

**SYSTEM:DATE****command/query**

Sets the date in the analyzer's battery-backed clock.

Command Syntax:           SYSTEM:DATE <year>, <month>, <day>

    <year> ::= [<number>|<bound>]  
    <number> ::= a real number (NRf data)  
                  limits: 0:9999  
    <bound> ::= MAX|MIN  
    <month> ::= [<number>|<bound>]  
    <number> ::= a real number (NRf data)  
                  limits: 1:12  
    <bound> ::= MAX|MIN  
    <day> ::= [<number>|<bound>]  
    <number> ::= a real number (NRf data)  
                  limits: 1:31  
    <bound> ::= MAX|MIN

Example Statements:        OUTPUT 711;"SYSTEM:DATE 1993, 9, 27"  
                            OUTPUT 711;"syst:date 1993, 12, 25"

Query Syntax:             SYSTEM:DATE?

Return Format:            Integer, Integer, Integer

Attribute Summary:        Option: not applicable  
                            Synchronization Required: no  
                            Preset State: not applicable  
                            SCPI Compliance: confirmed

Description:

You must enter the year as a four-digit number, including century and millennium information (1993, not 93).

**SYSTem:ERRor?**

**query**

Returns one error message from the analyzer's error queue.

Query Syntax:               SYSTem:ERRor?

Example Statements:        OUTPUT 711;"SYST:ERROR?"  
                              OUTPUT 711;"syst:error?"

Return Format:             Integer, "<STRING>"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not affected by Preset  
                              SCPI Compliance: confirmed

Description:

The error queue temporarily stores up to 5 error messages. When you send the SYST:ERR query, one message is moved from the error queue to the output queue so your controller can read the message. The error queue delivers messages to the output queue in the order received.

If more than 5 error messages are reported before any are read from the queue, the oldest error messages are saved. The last error message indicates too many error messages were received for the queue.

**Note**                    The error queue is cleared when you turn on the analyzer and when you send the \*CLS command.

**SYSTem:FAN[:STATe]****command/query**

Sets the fan's operating mode.

Command Syntax:           SYSTem:FAN[:STATe] OFF|0|FULL|AUTO

Example Statements:        OUTPUT 711;":Syst:Fan OFF"  
                              OUTPUT 711;"SYSTEM:FAN:STATE FULL"

Query Syntax:             SYSTem:FAN[:STATe]?

Return Format:             OFF|0|FULL|AUTO

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State:    AUTO  
                              SCPI Compliance: instrument-specific

**Description:**

This command allows you to change the operation of the analyzer's fan.

To operate the fan at maximum speed, send SYST:FAN FULL.

To operate the fan in relationship to the internal temperature of the analyzer, send SYST:FAN AUTO. The analyzer increases the speed of the fan as the internal temperature of the analyzer rises.

To turn off the fan, send SYST:FAN OFF. If the internal temperature of the analyzer rises to the point of damaging the internal circuitry, the analyzer turns on the fan and resets it to AUTO.

---

**SYSTem:FLOG:CLear****command**

Clears the fault log of all entries.

Command Syntax:           SYSTem:FLOG:CLear

Example Statements:        OUTPUT 711;"syst:flog:clear"  
                              OUTPUT 711;"Syst:Flog:Cle"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

The fault log lists any hardware failures. This command deletes all lines in the fault log.

If any test fails, contact your local Agilent Technologies Sales and Service Office or have a qualified service technician refer to the *Agilent 35670A Service Manual*.

**SYSTem:KEY****command/query**

Writes or queries front-panel key presses.

Command Syntax:           SYSTem:KEY <keycode>  
                              <keycode> ::= <number>|<step>|<bound>

Example Statements:        OUTPUT 711;"SYST:KEY 21"  
                              OUTPUT 711;"system:key 53"

Query Syntax:             SYSTem:KEY?

Return Format:             Integer

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

The query returns the keycode for the last key pressed. If the return value is 0, no key was pressed.

Sending the command with a keycode simulates pressing of that front-panel key. See table 21-1 for the front-panel keycodes.

\*RST clears the queue of keys.



Table 21-1. Front Panel Keycodes  
by Front Panel Keys

Scale	29
Disp Format	39
Analys	18
Trace Coord	23
Active Trace	4
Meas Data	2
Window	30
Freq	17
Inst Mode	1
Trigger	32
Source	22
Input	3
Avg	55
Pause/Cont	19
Start	6
System Utility	24
Disk Utility	33
Save/Recall	13
Help	54
Preset	31
Local/GPIB	45
Plot/Print	16
BASIC	35
F1	37
F2	40
F3	52
F4	14
F5	36
F6	21
F7	20
F8	53
F9	43
Rtn	38

Front Panel Keycodes  
by Front Panel Keys

0	47
1	51
2	15
3	44
4	49
5	11
6	41
7	48
8	12
9	46
. (decimal point)	9
+/-	42
Back Space	25
Marker Value	34
î	28
Å	26
Marker	50
Marker Fctn	10

Front Panel Keycodes  
by Keycode

1	Inst Mode
2	Meas Data
3	Input
4	Active Trace
6	Start
9	. (decimal point)
10	Marker Fctn
11	5
12	8
13	Save/Recall
14	F4
15	2
16	Plot/Print
17	Freq
18	Analys
19	Pause/Cont
20	F7
21	F6
22	Source
23	Trace Coord
24	System Utility
25	Back Space
26	⊕
28	[
29	Scale
30	Window
31	Preset
32	Trigger
33	Disk Utility
34	Marker Value
35	BASIC
36	F5
37	F1
38	Rtn
39	Disp Format
40	F2
41	6
42	+/-
43	F9
44	3
45	Local/GPIB

SYSTEM

Front Panel Keycodes  
by Keycode

46	9
47	0
48	7
49	4
50	Marker
51	1
52	F3
53	F8
54	Help
55	Avg

**SYSTem:KLOCK****command/query**

Disables the keyboard.

Command Syntax:           SYSTem:KLOCK OFF|0|ON|1

Example Statements:        OUTPUT 711;":SYST:KLOCK OFF"  
                              OUTPUT 711;"syst:klock ON"

Query Syntax:             SYSTem:KLOCK?

Return Format:             Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  OFF (+0)  
                              SCPI Compliance:  confirmed

Description:

This command allows your controller to disable the keyboard. This provides local lockout capability during the running of Instrument BASIC programs.

The query returns 1 if the keyboard is disabled.

**SYSTem:POWer:SOURce?**

**query**

Queries the setting of the rear-panel power select switch.

Query Syntax:               SYSTem:POWer:SOURce?

Example Statements:        OUTPUT 711;"system:power:source?"  
                               OUTPUT 711;"SYST:POW:SOUR?"

Return Format:              AC|DC

Attribute Summary:         Option:   not applicable  
                               Synchronization Required:  no  
                               Preset State:  not affected by Preset  
                               SCPI Compliance:  instrument-specific

---

**SYSTem:POWer:STATe****command/query**

Turns off the analyzer's power.

Command Syntax:           SYSTem:POWer:STATe OFF|0|ON|1

Example Statements:        OUTPUT 711;"syst:pow:stat off"  
                              OUTPUT 711;"System:Power:State Off"

Query Syntax:             SYSTem:POWer:STATe?

Return Format:            Integer

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

SYST:POW:STAT OFF turns off the analyzer. This allows you to turn off the analyzer within an Instrument BASIC program.

SYST:POW:STAT ON has no affect on the analyzer.

---

**SYSTem:PRESet**

**command**

Returns most of the analyzer's parameters to their preset states.\$Ipreset;device

Command Syntax:           SYSTem:PRESet

Example Statements:        OUTPUT 711;"Syst:Pres"  
                               OUTPUT 711;"SYSTEM:PRES"

Attribute Summary:         Option:   not applicable  
                               Synchronization Required: no  
                               Preset State: not applicable  
                               SCPI Compliance: confirmed

Description:

In addition to returning parameters to their preset states, this command does all of the following:

- Aborts any GPIB operations.
- Cancels any pending \*OPC command or query.
- Clears the error queue.
- Clears all event registers (sets all bits to 0).

The preset state of each parameter is listed under the Attribute Summary of the associated command.

---

**Note**                    This command is equivalent to the front panel **Preset** hardkey.

---

SYST:PRES does not affect the following parameters:

- the state of the Power-on Status Clear flag
- the state of all enable and transition registers
- the GPIB input and output queues
- the time and date (SYST:TIME and SYST:DATE)
- the GPIB address settings (SYST:COMM:GPIB:ADDR, HCOP:PLOT:ADDR, HCOP:PRIN:ADDR)
- the GPIB controller capability setting
- the default disk selection (MMEM:MSIS)
- the serial interface (RS-232-C) parameter settings (SYST:COMM:SERial)
- contents of limit, data, and waterfall registers
- contents of the math function and constant registers
- contents of the RAM disks
- calibration constants
- contents of the time capture buffer
- the external keyboard selection



**SYSTem:SET****command/query**

Transfers an instrument state between the analyzer and an external controller.

Command Syntax:           SYSTem:SET <STATE>

```

<STATE> ::= #<byte>[<length_bytes>]<data_bytes>
<byte>  ::= one byte specifying the number of length bytes
           to follow (ASCII-encoded)
<length_bytes> ::= bytes specifying the number of data bytes
                 to follow (ASCII-encoded)
<data_bytes> ::= the bytes that define an instrument state

```

Example Statements:       OUTPUT 711;":SYST:SET STATE"  
                           OUTPUT 711;"system:set state"

Query Syntax:            SYSTem:SET?

Return Format:            definite length <STATE>

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: not applicable  
                           SCPI Compliance: confirmed

## Description:

This command transfers a complete instrument state—the same information contained in a state file—between the analyzer and your controller. This allows you to store an instrument state on your controller's file system. The state cannot be altered.

When you transfer an instrument state to the analyzer, you can use either the definite or indefinite length block syntax. When the analyzer returns the state to a controller, it always uses the definite length block syntax. For more information about transferring block data, see the *GPIB Programmer's Guide*.

**SYSTem:TIME****command/query**

Sets the time in the analyzer's battery-backed clock.

Command Syntax:           SYSTem:TIME <hour>,<minute>,<second>

```
<hour> ::= [<number>|<bound>]
<number> ::= a real number (NRf data)
             limits: 0:23
<bound> ::= MAX|MIN
<minute> ::= [<number>|<bound>]
<number> ::= a real number (NRf data)
             limits: 0:59
<bound> ::= MAX|MIN
<second> ::= [<number>|<bound>]
<number> ::= a real number (NRf data)
             limits: 0:60
<bound> ::= MAX|MIN
```

Example Statements:       OUTPUT 711;"system:time 15, 5, 0"  
                          OUTPUT 711;"SYST:TIME 9, 30, 0"

Query Syntax:            SYSTem:TIME?

Return Format:            Integer, Integer, Integer

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

This command sets the time using a 24-hour format. For example, 3:05 pm becomes 15:05 and is sent as SYST:TIME 15, 5, 0.

---

**SYSTem:VERSion?****query**

Returns the SCPI version to which the analyzer complies.

Query Syntax:               SYSTem:VERSion?

Example Statements:        OUTPUT 711;":system:vers?"  
                              OUTPUT 711;"Syst:Version?"

Return Format:             Real

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

The format of the return is YYYY.V The Ys represent the SCPI year-version and the V represents the revision number for that year.



TEST

---



## TEST

---

Most of the commands in this subsystem are used to invoke service tests. Since these tests should be used only by qualified service personnel, the commands are not described here. See the Agilent 35670A Service Guide for descriptions.

Two commands in the TEST subsystem allow you to run the long confidence test and to determine whether the test passed or failed. These commands are described in this chapter.

---

**TEST:LOG:CLEar****command**

Clears the test log.

Command Syntax:           TEST:LOG:CLEar

Example Statements:        OUTPUT 711;"TEST:LOG:CLE"  
                              OUTPUT 711;"test:log:cle"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: instrument-specific

Description:

The test log lists the results of the long confidence test. This command deletes all lines in the test log.

If any test fails, contact your local Agilent Technologies Sales and Service Office or have a qualified service technician refer to the *Agilent 35670A Service Guide*.



---

**TEST:LONG****command**

Executes the long confidence test.

Command Syntax:           TEST:LONG

Example Statements:        OUTPUT 711;":Test:Long"  
                              OUTPUT 711;"TEST:LONG"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  not applicable  
                              SCPI Compliance:  instrument-specific

Description:

The long confidence test is a series of individual tests that check various analyzer functions.

The overall result of the long confidence test is available by sending TEST:LONG:RESult?.

If the long confidence test fails, contact your local Agilent Technologies Sales and Service Office or have a qualified service technician refer to the *Agilent 35670A Service Guide*.

---

**TEST:LONG:RESult?****query**

Returns the overall result of the long confidence test.

Query Syntax:                    TEST:LONG:RESult?

Example Statements:            OUTPUT 711;"test:long:res?"  
                                  OUTPUT 711;"Test:Long:Res?"

Return Format:                   Integer

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: instrument-specific

Description:

This query tells you whether or not the analyzer passed the last long confidence test. The query returns +0 if the analyzer failed, +1 if it passed.

To display the results of each test to the analyzer's screen, send the DISPLAY:VIEW TTAB command. To clear the test log, send TEST:LOG:CLEar. To display the fault log table (which lists hardware failures) send DISPLAY:VIEW FTAB. To clear the fault log, send SYSTem:FLOG:CLEar.

If the long confidence test fails, contact your local Agilent Technologies Sales and Service Office or have a qualified service technician refer to the *Agilent 35670A Service Guide*.

TRACe

---



## TRACe

---

This subsystem contains commands which provide access to the raw measurement data (data that has not been transformed into the current display coordinates). The commands, TRAC:DATA and TRAC:WAT:DATA, allow you to transfer measurement data between the analyzer and an external controller.

Figure 23-1 shows you the position of TRAC:DATA in the data flow. It also illustrates the difference between data available in the TRACe subsystem and the CALCulate subsystem.

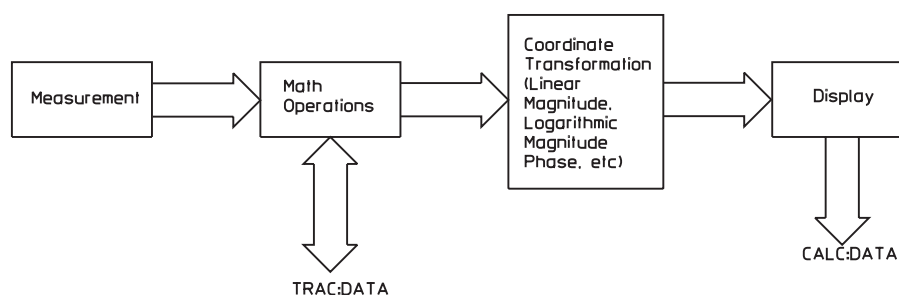


Figure 23-1. Flow of Measurement Data

After measurement data is collected, any specified math operations are performed. Data is then transformed into the specified coordinate system and sent to the display. TRAC:DATA gives you access to the raw measurement data after math operations have been performed. This data can be either complex or real. CALC:DATA gives you access to the display data—after the coordinate transformation.

---

**Note** Both TRAC:DATA and CALC:DATA allow you to transfer measurement data from the analyzer. Only TRAC:DATA, however, allows you to transfer measurement data to the analyzer.

---

**TRACe[:DATA]****command/query**

Stores data to the specified data register.

Command Syntax: TRACe[:DATA] {D1|D2|D3|D4|D5|D6|D7|D8}, &lt;DATA&gt;

```

<DATA> ::= D1|D2|D3|D4|D5|D6|D7|D8
        ::= TRACe1|TRACe2|TRACe3|TRACe4
        ::= CAL1|CAL2|CAL3|CAL4
           Calibration trace channel 1, 2, 3 or 4
        ::= <BLOCK> (definite and indefinite length)
        ::= <number>,<number>,<number> . . .
<number> ::= <integer>
          ::= <floating-point numeric>

```

```

Example Statements:  OUTPUT 711;"TRAC:DATA D1, TRAC2"
                   OUTPUT 711;"trac d3, d5"

```

See example program segments in Appendix F.

Query Syntax: TRACe[:DATA]? {D1|D2|D3|D4|D5|D6|D7|D8}

Return Format: &lt;DEF\_BLOCK&gt;

If FORMat[:DATA] REAL:

```

<DEF_BLOCK> ::= #<byte><length_bytes> <1st_value>[. . .<last_value>]
<byte> ::= one byte specifying the number of length bytes
           to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)

```

If FORMat[:DATA] ASCii:

```

<DEF_BLOCK> ::= <1st_value>[. . .<last_value>]

```

```

Attribute Summary:  Option: not applicable
                   Synchronization Required: no
                   Preset State: not applicable
                   SCPI Compliance: confirmed

```

## Description:

This command copies a selected trace into one of eight data registers, copies data between data registers, and replaces existing data in the specified data register with block data from the controller.

The first parameter specifies the destination. The second parameter specifies the data or the source of the data:

- one of the trace boxes; A (TRAC1), B (TRAC2), C (TRAC3), or D (TRAC4)
- a calibration trace for one of the input channels
- data from one of the data registers

See table 23-1 for the types of measurement data.

Table 23-1. Types of Data

CALC:FEED	Measurement Data	Type of Data
XTIM:CORR [1   2   3   4]	Auto correlation	Real
TCAP [1   2   3   4]	Capture buffer	Real
XFR:POW:COH	Coherence	Real
XFR:POW:COMP [1   2   3   4]	Composite Power	Real
XTIM:CORR:CROS [1,2   1,3   1,4   3,4]	Cross Correlation	Real
XFR:POW:CROS [1,2   1,3   1,4   3,4]	Cross Spectrum	Complex Pair (real,imaginary)
XTIM:VOLT:CDF [1   2   3   4]	Cumulative Density Function	Real
XFR:POW:RAT [2,1   3,1   4,1   4,3]	Frequency Response	Complex Pair (real,imaginary)
XTIM:VOLT:HIST [1   2   3   4]	Histogram	Real
XFR:POW:LIN [1   2   3   4]	Linear Spectrum	Complex Pair (real,imaginary)
XFR:POW:VAR [1   2   3   4]	Normalized Variance	Real
XVOL:VOLT [1,2   1,3   1,4   3,4]	Orbit Diagram	Real
XORD:TRACK [1   2   3   4],[1   2   3   4   5]	Order Track	Complex Pair (real,imaginary)
XFR:POW [1   2   3   4]	Power Spectrum	Real
XTIM:VOLT:PDF [1   2   3   4]	Probability Density Function	Real
XRPM:PROF	RPM Profile	Real
XTIM:VOLT [1   2   3   4]	Time	Real for baseband * Complex Pair (real,imaginary) for zoom
XTIM:VOLT [1   2   3   4]	Unfiltered Time	Real
XTIM:VOLT:WIND [1   2   3   4]	Windowed Time	Real for baseband * Complex Pair (real, imaginary)for zoom

\* Baseband measurements have a start frequency of 0 Hz; zoom measurements have a start frequency 0 Hz.

This command differs from the MMEM:STOR:TRAC command. The MMEM:STOR:TRAC command saves trace data to a file. TRAC:DATA saves trace data to one of the data registers.

The query form of this command transfers data from the analyzer over the bus to your controller.

---

**Note** Alias data is included in frequency domain data. Table 23-2 specifies which data points are alias-protected and which data points are not protected. The number of points is determined by the lines of resolution set with the [SENSE:]FREQUency:RESolution command.

---

Sending block data to a data register is valid only if the data register contains data. That is, you can only replace existing data in the data register. You cannot transfer block data to an empty data register. For more information about transferring block data, see Appendix F, “Example Programs,” or the  *GPIB Programmer’s Guide*.

Table 23-2. Alias-Protected Data by Lines of Resolution

FFT-Baseband Measurement Data Points

Lines of Resolution	Alias-Protected Data Points <sup>*</sup>	Data Points Not Protected <sup>*</sup>
100	0 to 100	101 to 128
200	0 to 200	201 to 256
400	0 to 400	401 to 512
800	0 to 800	801 to 1024

\* First point is Point 0.

FFT-Zoom (Start Frequency > 0 Hz)  
Measurement Data

Lines of Resolution	Alias-Protected Data Points <sup>*</sup>	Data Points <i>Not</i> Protected <sup>*</sup>
100	14 to 114	0 to 13 115 to 127
200	28 to 228	0 to 27 229 to 255
400	56 to 456	0 to 55 457 to 511
800	112 to 912	0 to 111 913 to 1023

\* First point is Point 0.



**TRACe:WATerfall[:DATA]****command/query**

Stores data to the specified waterfall register.

Command Syntax:           TRACe:WATerfall[:DATA] {W1|W2|W3|W4|W5|W6|W7|W8}, <WDATA>

```
<WDATA> ::= W1|W2|W3|W4|W5|W6|W7|W8
          ::= TRACe1|TRACe2|TRACe3|TRACe4
          ::= <BLOCK> (definite and indefinite length)
          ::= <number>,<number>,<number> . . .
<number> ::= <integer>
          ::= <floating-point numeric>
```

Example Statements:       OUTPUT 711;":TRACE:WAT W6, WDATA"  
                          OUTPUT 711;"trac:waterfall:data W7, WDATA"

Query Syntax:           TRACe:WATerfall[:DATA]?

Return Format:           <DEF\_BLOCK>

If FORMat[:DATA] REAL:

```
<DEF_BLOCK> ::= #<byte><length_bytes> <1st_axis>
               [. . .<last_value>]
<byte> ::= one byte specifying the number of length bytes
           to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)
```

If FORMat[:DATA] ASCii:

```
<DEF_BLOCK> ::= <1st_value>
               [. . .<last_value>]
```

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: not applicable  
                          SCPI Compliance: confirmed

Description:

This command copies a waterfall display into one of eight waterfall registers, transfers block data to a specified waterfall register and copies data between waterfall registers. The waterfall display must contain more than 1 trace.

The first parameter specifies the destination. The second parameter specifies the data or the source of the data:

- one of the trace boxes; A (TRAC1), B (TRAC2), C (TRAC3), or D (TRAC4)
- data to be loaded from one of the waterfall data registers

See table 23-1 on page 23-3 for the types of data.

## TRACe

This command differs from the MMEM:STOR:WAT command. The MMEM:STOR:WAT command saves a waterfall display to a file. TRAC:WAT:DATA saves a waterfall to one of the waterfall registers.

The query form of this command transfers data from the analyzer over the bus to your controller.

---

**Note** Alias data is included in frequency domain data. Table 23-2 (page 23-4) specifies which data points are alias-protected and which data points are not protected. The number of points is determined by the lines of resolution set with the [SENSe:]FREQuency:RESolution command.

---

For more information about transferring block data, see Appendix F, “Example Programs,” or the *GPiB Programmer’s Guide*.

**TRACe:X[:DATA]?****query**

Returns the X-axis data for trace displays.

Query Syntax:                   TRACe:X[:DATA]?  
 TRACe1 | TRACe2 | TRACe3 | TRACe4 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | W1 | W2  
 | W3 | W4 | W5 | W6 | W7 | W8

Example Statements:            OUTPUT 711;"Trace:X:Data? TRACE4"  
 OUTPUT 711;"TRACE:X? W3"

Return Format:                 <DEF\_BLOCK>

If FORMat[:DATA] REAL:

```
<DEF_BLOCK> ::= #<byte><length_bytes> <1st_axis>
               [. . .<last_value>]
<byte> ::= one byte specifying the number of length bytes
           to follow (ASCII encoded)
<length_bytes> ::= number of data bytes to follow (ASCII encoded)
```

If FORMat[:DATA] ASCii:

```
<DEF_BLOCK> ::= <1st_X- axis_value>
               [. . .<last_X-axis_value>]
```

Attribute Summary:            Option: not applicable  
 Synchronization Required: no  
 Preset State: not applicable  
 SCPI Compliance: instrument-specific

Description:

This query returns the values along the X-axis for any display. The values identify each bin in the trace.

---

**Note**                   Alias data is included in frequency domain data. Table 23-2 (page 23-44) specifies which data points are alias-protected and which data points are not protected. The number of points is determined by the lines of resolution set with the [SENSe:]FREQuency:RESolution command.

---

To determine the units for the X-axis send TRAC:X:UNIT?.

**TRACe:X:UNIT?****query**

Returns the unit for the x-axis for trace displays.

Query Syntax:                   TRACe:X:UNIT?  
                                  TRACe1 | TRACe2 | TRACe3 | TRACe4 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | W1 | W2  
                                  |  
                                  W3 | W4 | W5 | W6 | W7 | W8

Example Statements:            OUTPUT 711;"trace:x:unit? D2"

OUTPUT 711;"TRAC:X:UNIT? TRAC4"

Return Format:                 "<STRING>"

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: instrument-specific

Description:

The unit for the X-axis is dependent upon the type of measurement data selected.

To query the values along the X-axis, use TRAC:X?.

**TRACe:Z[:DATA]?****query**

Returns the Z-axis data for waterfall displays.

Query Syntax:                   TRACe:Z[:DATA]?  
 TRACe1 | TRACe2 | TRACe3 | TRACe4 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | W1 | W2  
 | W3 | W4 | W5 | W6 | W7 | W8

Example Statements:            OUTPUT 711;":trace:z? W3"  
 OUTPUT 711;"Trac:Z:Data? W5"

Return Format:                   <DEF\_BLOCK>

```

<DEF_BLOCK> ::= #<byte><length_bytes> <1st_axis>
               [. . .<last_value>]
  <byte> ::= one byte specifying the number of length bytes
            to follow (ASCII encoded)
  <length_bytes> ::= number of data bytes to follow (ASCII encoded)

```

Attribute Summary:            Option: not applicable  
 Synchronization Required: no  
 Preset State: not applicable  
 SCPI Compliance: instrument-specific

**Description:**

This query returns the values along the Z-axis in waterfall displays. The values identify each trace in a waterfall. It tells you when the measurement data was armed.

If the query is to trace box A (TRAC1), trace box B (TRAC2), trace box C (TRAC3), trace box D (TRAC4), or one of the waterfall registers, W1 - W8, the values are returned as an array.

---

**Note**                    The array returned from a waterfall register contains a value for each waterfall step. This can be a maximum of 32767 values. The number of values returned is equal to than the quantity specified with the CALC:WAT:COUN command. (Depending upon other factors, the value may be less than the quantity specified with CALC:WAT:COUN.)

---

If the query is to a data register (D1 - D8), a single value is returned. It is the Z-axis value of the trace saved to that data register.

To determine the units for the Z-axis, send TRAC:Z:UNIT?.

**TRACe:Z:UNIT?****query**

Returns the unit for the z-axis in waterfall displays.

Query Syntax: TRACe:Z:UNIT?  
 TRACe1 | TRACe2 | TRACe3 | TRACe4 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | W1 | W2  
 | W3 | W4 | W5 | W6 | W7 | W8

Example Statements: OUTPUT 711;"TRAC:Z:UNIT? W4"  
 OUTPUT 711;"trace:z:unit? D7"

Return Format: "COUNTS" | "AVG" | "SEC" | "RPM"

Attribute Summary: Option: not applicable  
 Synchronization Required: no  
 Preset State: not applicable  
 SCPI Compliance: instrument-specific

## Description:

The unit for the Z-axis is dependent upon the type of arming used to trigger the measurement.

The query returns "COUNTS" if manual trigger arming or automatic trigger arming was used. If the measurement data is averaged and manual or automatic arming was used, the query returns "AVG".

The query returns "SEC" if time step arming was used.

The query returns "RPM" if RPM step arming was used.

To query the values along the Z-axis, send TRAC:Z?.

TRIGger

---





# TRIGger

This subsystem contains commands that control the analyzer's triggering function. See the ARM subsystem for commands that control the trigger-arming functions. It also contains commands that configure the analyzer's tachometer.

Figure 24-1 shows the model for the Agilent 35670A's ARM-INITiate-TRIGger functions.

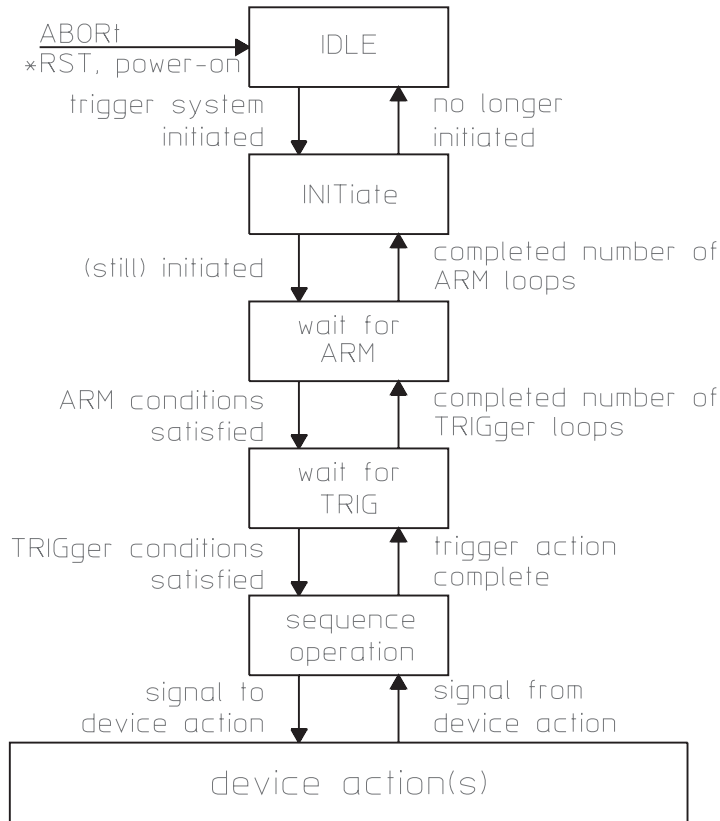


Figure 24-1. The Agilent 35670A's ARM-INIT-TRIG Functions

**TRIGger:EXTernal:FILTer[:LPAS][:STATe]****command/query**

Turns on a low pass filter for the external trigger.

Command Syntax: TRIGger:EXTernal:FILTer[:LPAS][:STATe] OFF|0|ON|1

Example Statements: OUTPUT 711;"TRIGGER:EXT:FILT:LPAS:STAT ON"  
OUTPUT 711;"trigger:ext:filt ON"

Query Syntax: TRIGger:EXTernal:FILTer[:LPAS][:STATe]?

Return Format: Integer

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: not applicable  
SCPI Compliance: instrument-specific

**Description:**

Use this command to improve the trigger signal from the external trigger input. The low pass filter eliminates noise.

See Online Help for additional information.

**TRIGger:EXTernal:LEVel****command/query**

Specifies the level of the external input signal which causes the analyzer to trigger.

Command Syntax: TRIGger:EXTernal:LEVel {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)

limits: -10.0:10.0

<unit> ::= [V]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;" :Trigger:Ext:Level -0.412237"  
OUTPUT 711;"TRIG:EXT:LEVEL 0.54171"

Query Syntax: TRIGger:EXTernal:LEVel?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: 0 V  
SCPI Compliance: instrument-specific

## Description:

Use this command to trigger the analyzer when the signal applied to the external trigger input connector passes through the specified value in the direction set by the TRIG:SLOP command. The external trigger connector is on the rear panel of the analyzer.

When TRIG:EXT:RANGE is set to HIGH, the allowable range is -10 V to +10 V. The level may be set to 78.125 mV increments. When TRIG:EXT:RANGE is set to LOW, the allowable range is -2V to +2V. The level may be set to 15.625 mV increments.

This command is only valid with external triggering (TRIG:SOUR EXT).

**TRIGger:EXTernal:RANGe****command/query**

Selects the range for the external trigger's level.

Command Syntax:           TRIGger:EXTernal:RANGe HIGH|LOW

Example Statements:        OUTPUT 711;"trig:external:rang HIGH"  
                              OUTPUT 711;"Trig:External:Rang LOW"

Query Syntax:             TRIGger:EXTernal:RANGe?

Return Format:             HIGH|LOW

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  HIGH  
                              SCPI Compliance:  instrument-specific

**Description:**

This command works with the TRIG:EXT:LEVEL command.

When TRIG:EXT:RANG is HIGH, the allowable range is -10 V to +10 V. The external trigger level may be set in 78.125 mV increments.

When TRIG:EXT:RANG is LOW, the allowable range is -2 V to + 2V. The external trigger level may be set in 15.625 mV increments.

This command is only valid with external triggering (TRIG:SOUR:EXT).

**TRIGger[:IMMEDIATE]****command**

Triggers the analyzer if TRIG:SOUR is BUS.

Command Syntax:           TRIGger[:IMMEDIATE]

Example Statements:        OUTPUT 711;":TRIGGER"  
                              OUTPUT 711;"trig:imm"

Attribute Summary:        Option:   not applicable  
                              Synchronization Required: no  
                              Preset State: not applicable  
                              SCPI Compliance: confirmed

Description:

This command triggers the analyzer if the following two conditions are met:

- The GPIB is designated as the trigger source. (Send the TRIG:SOUR:BUS command.)
- The analyzer is waiting to trigger. Bit 5 of the Operation Status condition register must be set.

The command is ignored at all other times.

The \*TRG command has the same effect as TRIG:IMM. It also has the same effect as the GPIB bus management command Group Execute Trigger (GET).

For more information about how to use status registers, see "Programming the Status System" in the *GPIB Programmer's Guide*.

**TRIGger:LEVel****command/query**

Specifies the level of the input signal which causes the analyzer to trigger.

Command Syntax:           TRIGger:LEVel {<number> [<unit>]} | <step> | <bound>

```
<number> ::= a real number (NRf data)
           limits: -9.9e37:9.9e37
<unit>   ::= [PCT|V|VPK|EU]
<step>   ::= UP|DOWN
<bound>  ::= MAX|MIN
```

Example Statements:       OUTPUT 711;"TRIGGER:LEV .10"  
                           OUTPUT 711;"trig:lev -50 PCT"  
                           OUTPUT 711;"TRIGGER:LEVEL 2.5 V"

Query Syntax:            TRIGger:LEVel?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                           Synchronization Required: no  
                           Preset State: +0 V  
                           SCPI Compliance: confirmed

## Description:

You can specify the trigger level as a value, as a percentage of the trigger channel's current input range, or as a fraction of the trigger channel's current input range. TRIG:LEV 0.25 is the same as TRIG:LEV 25 PCT.

To query the analyzer's current input range, send [SENSe:]VOLT[1|2|3|4]:RANG?. To determine the units, send VOLT[1|2|3|4]:RANG? UNIT.

The trigger source must be one of the analyzer's four input channels, TRIG:SOUR INT1, TRIG:SOUR INT2, TRIG:SOUR INT3, or TRIG:SOUR INT4.

**TRIGger:SLOPe****command/query**

Specifies the slope of the signal which triggers the analyzer.

Command Syntax:           TRIGger:SLOPe POSitive|NEGative

Example Statements:        OUTPUT 711;"Trigger:Slop POSITIVE"  
                              OUTPUT 711;"TRIGGER:SLOP NEGATIVE"

Query Syntax:             TRIGger:SLOPe?

Return Format:             POS|NEG

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  POS  
                              SCPI Compliance:  confirmed

**Description:**

The analyzer is triggered either by a low-to-high transition of the trigger signal (POS) or by the high-to-low transition of the trigger signal (NEG).

Use the TRIG:LEV or TRIG:EXT:LEV command to specify the trigger level.

This command is only valid with the following selections:

- TRIG:SOUR:EXT
- TRIG:SOUR:INT1
- TRIG:SOUR:INT2
- TRIG:SOUR:INT3
- TRIG:SOUR:INT4

The query returns the currently specified slope.

**TRIGger:SOURce****command/query**

Selects the source of the trigger event.

Command Syntax: TRIGger:SOURce IMMEDIATE|EXTER-  
nal|INTERNAL1|INTERNAL2|INTERNAL3|INTERNAL4|OUTPUT|BUS

Example Statements: OUTPUT 711;":trig:source INTERNAL2"  
OUTPUT 711;"Trig:Source BUS"

Query Syntax: TRIGger:SOURce?

Return Format: IMM|EXT|INT1|INT2|INT3|INT4|OUTP|BUS

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: IMM  
SCPI Compliance: confirmed

Description:

Triggering is not available in Swept Sine instrument mode (INST:SEL SINE).

To select free run triggering send IMM. The analyzer automatically triggers as soon as it is armed.

To select the analyzer's external trigger connector (on the rear panel) as the trigger source, send EXT. If EXT is selected, the analyzer is triggered when the signal applied to the external trigger input connector satisfies the conditions of TRIG:EXT:LEV and TRIG:SLOP. Low-to-high transition (TRIG:SLOP POS) and 0 V (TRIG:EXT:LEV) are the default values.

To select one of the analyzer's input channels as the trigger source, send INT[1|2|3|4]. If the input channel specifier is not sent, the selection defaults to Channel 1. The analyzer is triggered when the selected channel's input signal satisfies the conditions of TRIG:LEV and TRIG:SLOP.

To select the analyzer's signal source, send OUTP. The analyzer is triggered synchronously with the source.



To select the analyzer's GPIB connector (also on the rear panel) send BUS. The analyzer is triggered when you send any of the following GPIB commands:

- \*TRG
- TRIG:IMM
- Group Execute Trigger (GET)

GET is a bus management command. See "Command and Data Modes" in the *GPIB Programmer's Guide* for more information.

---

**Note**            The analyzer must be waiting to trigger when it receives an external trigger signal or bus trigger command, otherwise the signal or command is ignored.  
Bit 5 of the Operation Status condition register is set to 1 when the analyzer is waiting to trigger.

---

**TRIGger:STARt[1 | 2 | 3 | 4]****command/query**

Specifies a pre-trigger or a post-trigger delay for the specified channel.

Command Syntax: TRIGger:STARt [1 | 2 | 3 | 4] {<number> [<unit>]} | <step> | <bound>

<number> ::= a real number (NRf data)  
limits: -9.9e37:9.9e37

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements: OUTPUT 711;"TRIG:STAR2 -0.1"  
OUTPUT 711;"trigger:start 1.5"

Query Syntax: TRIGger:STARt [1 | 2 | 3 | 4] ?

Return Format: Real

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: +0  
SCPI Compliance: instrument-specific

## Description:

This command is valid in FFT, octave, correlation, and histogram instrument modes only. It is not valid in order analysis or swept sine instrument mode.

In FFT, correlation, and histogram instrument modes:

This command specifies the amount of time between two points: the point at which the analyzer is triggered and the point at which the specified channel starts collecting data.

Pre-trigger values are entered as a negative (–) quantity. The channel starts collecting data before the trigger point.

Post-trigger values are entered as a positive (+) quantity. The channel starts collecting data after the trigger point.

Pre-trigger and post-trigger values are specified in seconds, with resolution equal to 1 sample.

$$\text{maximum pre-trigger value} = 8192 \times \text{sample time}$$

$$\text{sample time}_{\text{FFT}} = \frac{1}{\text{frequencyspan} \times 2.56}$$

$$\text{sample time}_{\text{correlation}} = \text{record length} / 1024$$

$$\text{maximum post-trigger delay} = 2^{31} \times \text{sample time}$$

The maximum post-trigger delay cannot exceed 1 Msec ( $1 \times 10^6$  seconds). If the specified post-trigger delay is greater than 1 Msec, the analyzer uses 1 Msec.

The difference between the delay specified for the “earliest” channel and the delay specified for the “latest” channel is dependent upon frequency resolution (FREQ:RES) as shown in table 24-1. In correlation and histogram instrument modes, resolution is always 400 lines.

Table 24-1. Maximum Difference Between Delay Values

Resolution (lines)	Maximum Delay Difference (samples)	Maximum Delay Difference (time records)
800	6144	3
400	7168	7
200	7680	15
100	7936	31

In octave analysis instrument mode:

This command specifies the delay between triggering and the start of an octave measurement.

The maximum post-trigger delay is 99,999 seconds. The minimum is 0 seconds. Only a post-trigger delay, entered as a positive quantity (+), is valid. A pre-trigger delay is not available in octave measurements.

The post-trigger delay is applied to all channels. The channel specifier is not used in octave instrument mode.

**TRIGger:TACHometer:HOLDoff****command/query**

Specifies a time interval in which the tachometer trigger is inhibited.

Command Syntax:           TRIGger:TACHometer:HOLDoff {<number>[<unit>]}|<step>|<bound>

<number> ::= a real number (NRf data)  
          limits: 0.0:0.052224

<unit> ::= [S]

<step> ::= UP|DOWN

<bound> ::= MAX|MIN

Example Statements:       OUTPUT 711;"Trigger:tachometer:holdoff 0.05s"  
                          OUTPUT 711;"trig:tach:hold 2"

Query Syntax:            TRIGger:TACHometer:HOLDoff?

Return Format:            Real

Attribute Summary:       Option: not applicable  
                          Synchronization Required: no  
                          Preset State: +0.0  
                          SCPI Compliance: instrument-specific

**Description:**

This command causes the tachometer to ignore inputs for the specified time after each accepted tachometer event. It “holds off” the trigger for the tachometer for a specified amount of time.

See the ARM:SOURce commands for additional information about RPM step arming. See the [SENSe:]ORD commands for additional information about order tracking.



**TRIGger:TACHometer:PCOunt****command/query**

Specifies the number of tachometer pulses that occur in one revolution of the shaft.

Command Syntax:           TRIGger:TACHometer:PCOunt <number>|<step>|<bound>

<number> ::= a real number (NRf data)  
                  limits: 0.5:2048.0  
<step>    ::= UP|DOWN  
<bound>   ::= MAX|MIN

Example Statements:        OUTPUT 711;":TRIG:TACH:PCO 60"  
                              OUTPUT 711;"trigger:tach:pco 10"

Query Syntax:             TRIGger:TACHometer:PCOunt?

Return Format:             Real

Attribute Summary:        Option: not applicable  
                              Synchronization Required: no  
                              Preset State: +1.0  
                              SCPI Compliance: instrument-specific

Description:

See the ARM:SOURce commands for additional information about RPM step arming. See the [SENSe:]ORDER commands for additional information about order tracking.

---

**TRIGger:TACHometer:RANGe****command/query**

Specifies the input range of the analyzer's tachometer.

Command Syntax:           TRIGger:TACHometer:RANGe HIGH|LOW

Example Statements:        OUTPUT 711;"TRIG:TACH:RANGE HIGH"  
                              OUTPUT 711;"trig:tachometer:rang HIGH"

Query Syntax:             TRIGger:TACHometer:RANGe?

Return Format:             HIGH|LOW

Attribute Summary:        Option:   not applicable  
                              Synchronization Required:  no  
                              Preset State:  LOW  
                              SCPI Compliance:  instrument-specific

**Description:**

To specify the input range as  $\pm 4$  volts, send LOW.

To specify the input range as  $\pm 20$  volts, send HIGH.

See the ARM:SOURce commands for additional information about RPM step arming. See the [SENSe:]ORD commands for additional information about order tracking.

---

**TRIGger:TACHometer[:RPM]?****query**

Reads and returns the current RPM value for the analyzer's tachometer.

Query Syntax:                   TRIGger:TACHometer[:RPM]?

Example Statements:            OUTPUT 711;":Trig:Tachometer?"  
                                  OUTPUT 711;"TRIG:TACHOMETER:RPM?"

Return Format:                 Real

Attribute Summary:            Option:   not applicable  
                                  Synchronization Required: no  
                                  Preset State: not applicable  
                                  SCPI Compliance: instrument-specific

The tachometer must be turned on before it can be read. The tachometer is always on in the order analysis modes; in other modes, use the DISP:RPM ON command to turn it on.



**TRIGger:TACHometer:SLOPe****command/query**

Specifies the slope of the tachometer input signal to be used in RPM step arming or order tracking.

Command Syntax: TRIGger:TACHometer:SLOPe POSitive|NEGative

Example Statements: OUTPUT 711;"trig:tachometer:slop NEGATIVE"  
OUTPUT 711;"Trigger:Tach:Slop NEGATIVE"

Query Syntax: TRIGger:TACHometer:SLOPe?

Return Format: POS|NEG

Attribute Summary: Option: not applicable  
Synchronization Required: no  
Preset State: POS  
SCPI Compliance: instrument-specific

**Description:**

To specify a rising slope (low-to-high transition), send TRIG:TACH:SLOP POS.

To specify a falling slope (high-to-low transition), send TRIG:TACH:SLOP NEG.

The query returns the currently-specified slope.

See the ARM:SOURce commands for additional information about RPM step arming. See the [SENSe:]ORD commands for additional information about order tracking.



## Agilent 35670A Command Summary

---



## Agilent 35670A Command Summary

---

### Introduction

This appendix contains of all the GPIB commands recognized by the Agilent 35670A and a brief description. All commands have a query form unless specified as command only or query only.

The appendix lists common commands and then lists the subsystem commands in alphabetical order.

### Command List

Command Commands	Form	Description
*CAL?	query only	Calibrates the analyzer and returns the result
*CLS	command only	Clears the Status Byte by emptying the error queue and clearing all event registers
*ESE	‡	Sets bits in the Standard Event enable register
*ESR?	query only	Reads and clears the Standard Event event register
*IDN?	query only	Returns a string that uniquely identifies the analyzer
*OPC	‡	Sets or queries completion of all pending overlapped commands
*OPT?	query only	Returns a string that identifies the analyzer's option configuration
*PCB	command only	Sets the pass-control-back address
*PSC	‡	Sets the state of the Power-on Status Clear flag
*RST	command only	Executes a device reset
*SRE	‡	Sets bits in the Service Request enable register
*STB?	query only	Reads the Status Byte register
*TRG	command only	Triggers the analyzer when TRIG:SOUR is BUS
*TST?	query only	Tests the analyzer hardware and returns the results
*WAI	command only	Holds off processing of subsequent commands until all preceding commands have been processed

‡ Command form; add ``?' for query form

Subsystem Commands	Form	Description
ABORt	command only	Stops the current measurement in progress
<b>ARM</b>		
ARM[:IMMediate]	command only	Arms the trigger if ARM:SOUR is MAN
ARM:RPM:INCRement	‡	Specifies the number of RPM in a step for RPM step arming
ARM:RPM:MODE	‡	Enables the Start RPM Arming qualifier
ARM:RPM:THReshold	‡	Specifies the starting RPM value
ARM:SOURce	‡	Specifies the type of arming for the analyzer's trigger
ARM:TIMer	‡	Specifies the size of the step used in time step arming
<b>CALCulate</b>		
CALCulate[1   2   3   4]:ACTive	‡	Selects the active trace(s)
CALC[1   2   3   4]:CFIT:ABORt	command only	Aborts the curve fit operation
CALC[1   2   3   4]:CFIT:COpy	command only	Copies the synthesis table into the curve-fit table
CALC[1   2   3   4]:CFIT:DATA	‡	Loads values into the curve fit table
CALC[1   2   3   4]:CFIT:DESTination	‡	Selects the data register for the results of the curve fit operation
CALC[1   2   3   4]:CFIT:FREQuency[:AUTO]	‡	Specifies the region included in the curve fit operation
CALC[1   2   3   4]:CFIT:FREQuency:STARt	‡	Specifies the start frequency for a curve fit operation over a limited frequency span
CALC[1   2   3   4]:CFIT:FREQuency:STOP	‡	Specifies the stop frequency for a curve fit operation over a limited frequency span
CALC[1   2   3   4]:CFIT:FSCale	‡	Specifies the frequency scaling used in the curve fit operation
CALC[1   2   3   4]:CFIT[:IMMediate]	command only	Starts the curve fit process
CALC[1   2   3   4]:CFIT:ORDer:AUTO	‡	Determines the operation of the curve fitter
CALC[1   2   3   4]:CFIT:ORDer:POLes	‡	Specifies the number of poles used in the curve fit operation
CALC[1   2   3   4]:CFIT:ORDer:ZERos	‡	Specifies the number of zeros used in the curve fit operation
CALC[1   2   3   4]:CFIT:TDELay	‡	Specifies a time delay value for the curve fit operation
CALC[1   2   3   4]:CFIT:WEIGHt:AUTO	‡	Determines the weighting function used in the curve fit operation
CALC[1   2   3   4]:CFIT:WEIGHt:REGister	‡	Selects the data register which contains the weighting function for the curve fit operation
CALC[1   2   3   4]:DATA?	query only	Returns trace data that has been transformed to the currently selected coordinate transform (specified with the CALC:FORMat command)
CALC[1   2   3   4]:DATA:HEADer:POINts?	query only	Returns the number of values in the data block returned with the CALC:DATA? query
CALC[1   2   3   4]:FEED	‡	Selects the measurement data to be displayed in the specified trace

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
CALC[1   2   3   4]:FORMat	‡	Selects a coordinate system for displaying measurement data and for transferring coordinate transformed data to a controller
CALC[1   2   3   4]:GDAPerture:APERture	‡	Specifies the phase-smoothing aperture used with group delay trace coordinates
CALC[1   2   3   4]:LIMit:BEEP[:STATe]	‡	Turns the limit-fail beeper on and off
CALC[1   2   3   4]:LIMit:FAIL?	query only	Returns the result of the last limit test; 0 for pass or 1 for fail
CALC[1   2   3   4]:LIM:LOW:CLEar[:IMMEDIATE]	command only	Deletes the lower limit line from the specified display
CALC[1   2   3   4]:LIMit:LOWer:MOVE:Y	command only	Moves all segments of the lower limit line up or down in the specified trace
CALC[1   2   3   4]:LIMit:LOWer:REPort[:DATA]?	query only	Returns the X-axis value of the failed points for the lower limit test
CALC[1   2   3   4]:LIMit:LOWer:REPort:YDATA?	query only	Returns the Y-axis value of the failed points for the lower limit test
CALC[1   2   3   4]:LIMit:LOWer:SEGment	‡	Defines the lower limit as a series of line segments in the specified display
CALC[1   2   3   4]:LIMit:LOWer:SEGment:CLEar	command only	Deletes a segment from the lower limit line
CALC[1   2   3   4]:LIM:LOW:TRACe[:IMMEDIATE]	command only	Converts the specified trace into a lower limit line
CALC[1   2   3   4]:LIMit:STATe	‡	Turns limit testing on and off for the specified trace
CALC[1   2   3   4]:LIM:UPPer:CLEar[:IMMEDIATE]	command only	Deletes the upper limit line from the specified display
CALC[1   2   3   4]:LIMit:UPPer:MOVE:Y	command only	Moves all segments of the upper limit line up or down in the specified trace box
CALC[1   2   3   4]:LIMit:UPPer:REPort[:DATA]?	query only	Returns the X-axis value of the failed points for the upper limit test
CALC[1   2   3   4]:LIMit:UPPer:REPort:YDATA?	query only	Returns the Y-axis value of the failed points for the upper limit test
CALC[1   2   3   4]:LIMit:UPPer:SEGment	‡	Defines the upper limit as a series of line segments in the specified display
CALC[1   2   3   4]:LIMit:UPPer:SEGment:CLEar	command only	Deletes a segment from the upper limit line
CALC[1   2   3   4]:LIM:UPP:TRACe[:IMMEDIATE]	command only	Converts the specified trace into an upper limit line
CALC[1   2   3   4]:MARKer:BAND:START	‡	Specifies the lowest frequency of the band in which power is calculated
CALC[1   2   3   4]:MARKer:BAND:STOP	‡	Specifies the highest frequency of the band in which power is calculated
CALC[1   2   3   4]:MARKer:COUPled[:STATe]	‡	Couples the markers on all traces with the marker of the primary active trace
CALC[1   2   3   4]:MARK:DTABLE:CLEar[:IMM]	command only	Clears all values in the specified data table
CALC[1   2   3   4]:MARK:DTABl:COPIY[1   2   3   4]	command only	Copies the data table from one trace to another trace
CALC[1   2   3   4]:MARKer:DTABl[:DATA]?	query only	Returns the dependent values in the specified data table

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
CALC[1   2   3   4]:MARKer:DTABle:X[:DATA]?	query only	Returns the X values in the specified data table
CALC[1   2   3   4]:MARKer:DTABle:X:DELeTe	command only	Deletes the selected entry in the data table
CALC[1   2   3   4]:MARKer:DTABle:X:INSert	‡	Inserts an entry into the data table
CALC[1   2   3   4]:MARKer:DTABle:X:LABel	‡	Loads a label for the selected entry in the data table
CALC[1   2   3   4]:MARK:DTABl:X:SELeCt[:POINt]	‡	Selects the data table entry
CALC[1   2   3   4]:MARKer:FUNCTion	‡	Selects one of the analyzer's marker functions
CALC[1   2   3   4]:MARKer:FUNCTion:RESult?	query only	Returns the result of the calculation for the currently selected marker function
CALC[1   2   3   4]:MARKer:HARMonic:COUNT	‡	Specifies the maximum number of harmonic markers for the display
CALC[1   2   3   4]:MARK:HARM:FUNDamental	‡	Specifies the fundamental frequency for harmonic markers and calculations
CALC[1   2   3   4]:MARKer:MAXimum[:GLOBal]	command only	Moves the main marker to the highest point on the specified trace
CALC[1   2   3   4]:MARK:MAXi[:GLOBal]:TRACk	‡	Turns the peak tracking function on or off
CALC[1   2   3   4]:MARKer:MAXimum:LEFT	command only	Moves the main marker one peak to the left of its current location on the specified trace
CALC[1   2   3   4]:MARKer:MAXimum:RIGHT	command only	Moves the main marker one peak to the right of its current location on the specified trace
CALC[1   2   3   4]:MARKer:MODE	‡	Selects absolute or relative marker values
CALC[1   2   3   4]:MARKer:POSition	‡	Specifies the main marker's independent axis position
CALC[1   2   3   4]:MARKer:POSition:POINt	‡	Moves the main marker to a specific display point
CALC[1   2   3   4]:MARKer:REFerence:X	‡	Specifies the marker reference X-axis position
CALC[1   2   3   4]:MARKer:REFerence:Y	‡	Specifies the marker reference Y-axis position
CALC[1   2   3   4]:MARKer:SIDeband:CARRier	‡	Specifies the carrier frequency used for sideband markers and calculations
CALC[1   2   3   4]:MARKer:SIDeband:COUNT	‡	Specifies the number of sideband markers for the display
CALC[1   2   3   4]:MARKer:SIDeband:INCRement	‡	Specifies the frequency increment (or delta) between sideband markers
CALC[1   2   3   4]:MARKer[:STATe]	‡	Turns on the main markers or turns off all markers and marker functions for a selected trace
CALC[1   2   3   4]:MARKer:X[:ABSolute]	‡	Specifies the main marker's X-axis position
CALC[1   2   3   4]:MARKer:X:RELative	‡	Specifies the marker reference X-axis position relative to the main marker
CALC[1   2   3   4]:MARKer:Y[:ABSolute]?	query only	Returns the main marker's Y-axis position
CALC[1   2   3   4]:MARKer:Y:RELative	‡	Specifies the marker reference Y-axis position relative to the main marker
CALC[1   2   3   4]:MATH:CONStant[1   2   3   4   5]	‡	Defines the value of one of the constant registers

‡ Command form; add "?" for query form



Subsystem Commands	Form	Description
CALC[1   2   3   4]:MATH:DATA	‡	Loads a complete set of math definitions
CALC[1   2   3   4]:MATH:EXPR[1   2   3   4   5]]	‡	Defines a math function
CALC[1   2   3   4]:MATH:SElect	‡	Selects and displays the designated math function
CALC[1   2   3   4]:MATH:STATe	‡	Evaluates the currently selected math operation for the specified trace and displays the results
CALC[1   2   3   4]:SYNThesis:COpy	command only	Copies the contents of the curve fit table into the synthesis table
CALC[1   2   3   4]:SYNThesis:DATA	‡	Loads values into the synthesis table
CALC[1   2   3   4]:SYNThesis:DESTination	‡	Selects the data register for the results of the synthesis operation
CALC[1   2   3   4]:SYNThesis:FSCale	‡	Specifies a frequency scale for the synthesis operation
CALC[1   2   3   4]:SYNThesis:GAIN	‡	Specifies the gain constant
CALC[1   2   3   4]:SYNThesis[:IMMEDIATE]	command only	Creates a frequency response curve from the synthesis table
CALC[1   2   3   4]:SYNThesis:SPACing	‡	Specifies a linear or logarithmic scale for the X-axis data spacing
CALC[1   2   3   4]:SYNThesis:TDELay	‡	Specifies a time delay value for the synthesis operation
CALC[1   2   3   4]:SYNThesis:TTPe	‡	Converts the synthesis table to another table format
CALC[1   2   3   4]:UNIT:AMPLitude	‡	Selects the unit of amplitude for the Y-axis scale
CALC[1   2   3   4]:UNIT:ANGLe	‡	Specifies the unit for phase coordinates
CALC[1   2   3   4]:UNIT:DBReference	‡	Specifies the reference for dB magnitude trace coordinates
CALC[1   2   3   4]:UNIT:DBReference:IMPedance	‡	Specifies the system's reference impedance value in ohms ( $\Omega$ )
CALC[1   2   3   4]:UNIT:DBR:USER:LABel	‡	Assigns a name to the Y-axis unit when CALC:UNIT:DBR 'USER'
CALC[1   2   3   4]:UNIT:DBR:USER:REFerence	‡	Specifies the reference level for CALC:UNIT:DBR 'USER'
CALC[1   2   3   4]:UNIT:MECHanical	‡	Converts the Y-axis trace coordinates from the input's transducer units to the selected engineering units on the display
CALC[1   2   3   4]:UNIT:VOLTage	‡	Selects the vertical unit for the display's Y-axis
CALC[1   2   3   4]:UNIT:X	‡	Specifies the X-axis unit
CALC[1   2   3   4]:UNIT:X:ORDER:FACTor	‡	Specifies the speed of rotation in Hertz per Order or RPM per Order
CALC[1   2   3   4]:UNIT:X:USER:FREQ:FACTor	‡	Specifies the frequency conversion factor for user-defined X-axis units
CALC[1   2   3   4]:UNIT:X:USER:FREQ:LABel	‡	Assigns a name to the user-defined X-axis units in the frequency domain
CALC[1   2   3   4]:UNIT:X:USER:TIME:FACTor	‡	Specifies the time conversion factor for user-defined X-axis units

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
CALC[1   2   3   4]:UNIT:X:USER:TIME:LABel	‡	Assigns a name to the user-defined X-axis units in the time domain
CALC[1   2   3   4]:WATerfall:COUNT	‡	Specifies the number of traces stored for waterfall displays
CALC[1   2   3   4]:WATerfall[:DATA]?	query only	Returns waterfall data that has been transformed to the currently selected coordinate transform (specified with CALC:FORMat)
CALC[1   2   3   4]:WATerfall:SLICE:COPY	command only	Copies the selected waterfall slice to the designated data register
CALC[1   2   3   4]:WATerfall:SLICE:SElect	‡	Selects the waterfall slice at the specified X-axis position
CALC[1   2   3   4]:WATerfall:SLICE:SElect:POINt	‡	Selects a waterfall slice by its display point value
CALC[1   2   3   4]:WATerfall:TRACe:COPY	command only	Saves the trace to the specified data register
CALC[1   2   3   4]:WATerfall:TRACe:SElect	‡	Selects a waterfall trace by its Z-axis value
CALC[1   2   3   4]:WAT:TRACe:SElect:POINt	‡	Selects a waterfall trace by its step value
CALC[1   2   3   4]:X:DATA?	query only	Returns X-axis values that correspond to the Y-axis values read with the CALC:DATA? query
<b>CALibration</b>		
CALibration[:ALL]?	query only	Calibrates the analyzer and returns the result
CALibration:AUTO	‡	Calibrates the analyzer or sets the state of the autocalibration function
<b>DISPlay</b>		
DISPlay:ANNOtation[:ALL]	‡	Turns the display of screen annotation on or off
DISPlay:BODE	command only	Displays a Bode diagram
DISPlay:BRIGhtness	‡	Adjusts the intensity of the display
DISPlay:ERRor	command only	Displays text in the same format as the analyzer
DISPlay:EXTernal[:STATe]	‡	Enables the use of an external monitor
DISPlay:FORMat	‡	Selects a format for displaying trace data
DISPlay:GPIB:ECHO	‡	Enables and disables the echoing of GPIB command mnemonics to the analyzer's screen
DISPlay:PROGram:KEY:BOX	‡	Draws a box around softkey in an Instrument BASIC program
DISPlay:PROGram:KEY:BRACket	‡	Draws a bracket around one or more softkeys in an Instrument BASIC program
DISPlay:PROGram[:MODE]	‡	Selects the portion of the analyzer's screen to be used for Instrument BASIC program output
DISPlay:PROGram:VECTor:BUFFer[:STATe]	‡	Enables or disables the buffering of lines drawn with Instrument BASIC's graphics statements

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
DISPlay:RPM[:STATe]	‡	Turns on the display of the RPM indicator
DISPlay:STATe	‡	Turns on (or turns off) the analyzer's display
DISPlay:TCAPture:ENVELOpe[:STATe]	‡	Displays the <i>envelope</i> of the time capture buffer
DISPlay:VIEW	‡	Specifies what is displayed on the analyzer's screen
DISP[:WIND[1   2   3   4]]:DTAB:MARKer[:STATe]	‡	Turns on or off the markers for data tables
DISPlay[:WINDow[1   2   3   4]]:DTABle[:STATe]	‡	Enables and displays a data table
DISPlay[:WINDow[1   2   3   4]]:LIMit:STATe	‡	Turns limit lines on or off in the specified display
DISP[:WINDow[1   2   3   4]]:POLar:CLOCKwise	‡	Specifies the direction of the vector in a polar diagram
DISPlay[:WINDow[1   2   3   4]]:POLar:ROTation	‡	Specifies the angle of rotation for the polar diagram
DISP[:WIND[1   2   3   4]]:TRAC:APOWer[:STATe]	‡	Turns on or off the display of the A-weight total power band
DISP[:WIND[1   2   3   4]]:TRAC:BPOW[:STATe]	‡	Turns on or off the display of the total power band
DISP[:WIND[1   2   3   4]]:TRACe:GRATICule:GRID[:STATe]	‡	Turns on or off the display's overlay grid
DISPlay[:WINDow[1   2   3   4]]:TRACe:LABel	‡	Loads a label for the specified trace
DISP[:WIND[1   2   3   4]]:TRAC:LAB:DEF[:STAT]	‡	Turns on or off the analyzer's default title for the specified trace
DISP[:WIND[1   2   3   4]]:TRAC:X:MATC[1   2   3   4]	command only	Modifies the X-axis scaling of a trace to match the X-axis scaling of the reference trace
DISP[:WIND[1   2   3   4]]:TRAC:X[:SCAL]:AUTO	‡	Scales the measurement data to fit the trace box
DISP[:WIND[1   2   3   4]]:TRAC:X[:SCAL]:LEFT	‡	Specifies the first X-axis value on the display
DISP[:WIND[1   2   3   4]]:TRAC:X[:SCAL]:RIGHT	‡	Specifies the last X-axis value on the display
DISP[:WINDow[1   2   3   4]]:TRACe:X:SPACing	‡	Specifies X-axis scaling
DISP[:WIND[1   2   3   4]]:TRAC:Y:MATCh[1   2   3   4]	command only	Modifies the Y-axis scaling of a trace to match the Y-axis scaling of the reference trace
DISP[:WIND[1   2   3   4]]:TRAC:Y[:SCAL]:AUTO	‡	Scales and repositions the trace vertically to provide the best display of trace data
DISP[:WIND[1   2   3   4]]:TRAC:Y[:SCAL]:BOTT	‡	Specifies the value of the bottom reference point of the display's Y-axis scale
DISP[:WIND[1   2   3   4]]:TRAC:Y[:SCAL]:CENT	‡	Specifies the value of the center reference point of the display's Y-axis scale
DISP[:WIND[1   2   3   4]]:TRAC:Y[:SCAL]:PDIV	‡	Defines the height of each vertical division on the specified trace
DISP[:WIND[1   2   3   4]]:TRAC:Y[:SCAL]:REF	‡	Determines the Y-axis reference position for the specified display

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
DISP[:WIND[1   2   3   4]]:TRACe:Y:SCALe]:TOP	‡	Specifies the value of the top reference point of the display's Y-axis scale
DISP[:WINDow[1   2   3   4]]:TRACe:Y:SPACing	‡	Specifies scaling of the Y-axis for linear magnitude coordinate data
DISP[:WINDow[1   2   3   4]]:WATERfall:BASEline	‡	Specifies the percentage of each trace that is concealed in the waterfall display
DISP[:WINDow[1   2   3   4]]:WATERfall:BOTTom	‡	Specifies the bottom Z-axis value in a waterfall display
DISPlay[:WINDow[1   2   3   4]]:WATERfall:COUNT	‡	Determines the number of traces displayed in the waterfall trace box
DISP[:WINDow[1   2   3   4]]:WATERfall:HEIGht	‡	Specifies the height of a waterfall trace box
DISP[:WINDow[1   2   3   4]]:WATERfall:HIDDEN	‡	Turns on or off the removal of hidden waterfall traces
DISPlay[:WINDow[1   2   3   4]]:WATERfall:SKEW	‡	Enables a skewed waterfall display
DISP[:WIND[1   2   3   4]]:WATERfall:SKEW:ANGLE	‡	Specifies the amount of horizontal offset for waterfall displays
DISP[:WINDow[1   2   3   4]]:WATERfall[:STATe]	‡	Turns on or off the waterfall display for the specified trace box
DISPlay[:WINDow[1   2   3   4]]:WATERfall:TOP	‡	Specifies the top Z-axis value in a waterfall display
<b>FORMat</b>		
FORMat[:DATA]	‡	Specifies the data type and date encoding to be used during transfers of a data block
<b>HCOPy</b>		
HCOPy:COLor:DEFault	command only	Specifies default values for the plotter pen assignments
HCOPy:DESTination	‡	Specifies where the print or plot operation is sent: either to a device or to a file on the default disk
HCOPy:DEVIce:LANGUage	‡	Specifies the format of the plot/print output
HCOPy:DEVIce:SPEEd	‡	Specifies the plotting speed for all plotting operations initiated by the analyzer
HCOPy[:IMMEdiate]	command only	Plots or prints the currently specified item
HCOPy:ITEM:ALL[:IMMEdiate]	command only	Plots or prints the entire screen
HCOPy:ITEM:FFEEd:STATe	‡	Turns the page-eject feature on or off
HCOPy:ITEM:LABel:COLor	‡	Selects the pen used for plotting miscellaneous annotations
HCOPy:ITEM:LABel:STATe	‡	Prints a label on the plot and print output
HCOPy:ITEM:LABel:TEXT	‡	Specifies a label for plot and print output
HCOPy:ITEM:TDSTamp:FORMat	‡	Specifies the format of the time stamp used for plotting and printing
HCOPy:ITEM:TDSTamp:STATe	‡	Turns a time stamp on or off for print and plot operations
HCOP:ITEM[:WINDow[1   2   3   4]]:TRACe:COLor	‡	Selects the pen used to plot the specified trace and annotation

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe		
:GRATicule:COLor	‡	Selects the pen used to plot the trace graticules
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe		
:GRATicule[:IMMediate]	command only	Plots or prints all trace graticules
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe[:IMMediate]	command only	Plots or prints the currently displayed trace(s)
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe:LIMit:LTYPe	‡	Selects the line type for the specified limit line
HCOP:ITEM[:WIND[1   2   3   4]]:TRACe:LTYPe	‡	Selects the line type for the specified trace
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe:MARKer:COLo r	‡	Selects the pen used to plot markers for the specified trace
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe		
:MARKer[:IMMediate]	command only	Plots or prints the all currently displayed markers
HCOPy:ITEM[:WINDow[1   2   3   4]]:TRACe		
:MARKer:REFerence[:IMMediate]	command only	Plots or prints the marker reference for the currently displayed trace(s)
HCOPy:PAGE:DIMensions:AUTO	‡	Specifies P1 and P2 values for a plotter
HCOPy:PAGE:DIMensions:USER:LLEFt	‡	Specifies the lower left position (P1) of the plot area
HCOPy:PAGE:DIMensions:USER:URIGHt	‡	Specifies the upper right position (P2) of the plot area
HCOPy:PLOT:ADDRess	‡	Tells the analyzer which GPIB address is assigned to your plotter
HCOPy:PRINt:ADDRess	‡	Tells the analyzer which GPIB address is assigned to your printer
HCOPy:TITLe[1   2]	‡	Specifies a label for plot and print output
<b>INITiate</b>		
INITiate:CONTInuous	‡	Sets the trigger system to a continuously initiated state
INITiate[:IMMediate]	command only	Starts a measurement and forces the trigger system to exit the idle state
<b>INPut</b>		
INPut[1   2   3   4]:BIAS[:STATe]	‡	Enables/disables the ICP supply on the corresponding input channel
INPut[1   2   3   4]:COUPLing	‡	Selects AC or DC coupling for the specified channel
INPut[1   2   3   4]:FILTer:AWEighting[:STATe]	‡	Enables/disables the A-weight filter on the specified input channel
INPut[1   2   3   4]:FILTer[:LPASs][:STATe]	‡	Enables/disables the anti-alias filter for the specified input channel
INPut[1   2   3   4]:LOW	‡	Sets the specified channel's input shield to float or to ground

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
INPut[1   2   3   4]:REFErence:DIRection	‡	Sets the direction for the transducer point
INPut[1   2   3   4]:REFErence:POINt	‡	Sets the number for the transducer point
INPut[1   2   3   4]:STATe]	‡	Specifies one-channel
<b>INSTRument</b>		
INSTRument:NSElect	‡	Selects one of the analyzer's six major instrument modes
INSTRument[:SElect]	‡	Selects one of the analyzer's six major instrument modes
<b>MEMory</b>		
MEMory:CATalog[:ALL]?	query only	Returns information on the current contents and state of the analyzer's memory
MEMory:CATalog:NAME?	query only	Returns information about memory usage allocated for a specific item
MEMory:DELeTe:ALL	command only	Purges all allocated memory in the analyzer
MEMory:DELeTe[:NAME]	command only	Purges the memory allocated for a specific item
MEMory:FREE[:ALL]?	query only	Returns information on the state of the analyzer's memory
<b>MMEMory</b>		
MMEMory:COpy	command only	Copies the contents of one disk to another or one file to another
MMEMory:DELeTe	command only	Deletes one file or the contents of an entire disk
MMEMory:DISK:ADDRess	‡	Tells the analyzer which GPIB address is assigned to your external disk
MMEMory:DISK:UNIT	‡	Specifies the unit of the external disk drive
MMEMory:FSYStem?	query only	Returns the type of file system for the default disk
MMEMory:INITialize	command only	Formats the specified disk
MMEMory:LOAD:CFIT	command only	Loads a curve fit table into the analyzer from a file on the specified disk
MMEMory:LOAD:CONTInue	‡	Continues the load operation of time capture and waterfall files saved on multiple disks
MMEMory:LOAD:DTABle:TRACe[1   2   3   4]	command only	Loads a data table into the analyzer from a file on the specified disk
MMEMory:LOAD:LIMit:LOWer:TRACe[1   2   3   4]	command only	Loads a lower limit for the specified trace from a file on the specified disk
MMEMory:LOAD:LIMit:UPPer:TRACe[1   2   3   4]	command only	Loads an upper limit for the specified trace from a file on the specified disk

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
MMEMory:LOAD:MATH	command only	Loads a complete set of math definitions into the analyzer from the specified disk
MMEMory:LOAD:PROGram	command only	Loads an Instrument BASIC program into the analyzer from a file on the specified disk
MMEMory:LOAD:STATe	command only	Loads an instrument state into the analyzer from the specified disk
MMEMory:LOAD:SYNTHeSis	command only	Loads a synthesis table into the analyzer from a file on the specified disk
MMEMory:LOAD:TCAPtUre	command only	Loads a time capture file from the specified disk
MMEMory:LOAD:TRACe	command only	Loads a trace into the analyzer from the specified disk
MMEMory:LOAD:WATERfall	command only	Loads a waterfall file into the analyzer from a file on the specified disk
MMEMory:MDIRectory	command only	@RSUB1 – Command Syntax:
MMEMory:MOVE	command only	Renames a file
MMEMory:MSIS	‡	Specifies a default disk
MMEMory:NAME	‡	Specifies a filename for the output of a print or plot operation
MMEMory:STORe:CFIT	command only	Stores a curve fit table to a file on the specified disk
MMEMory:STORe:CONTInue	‡	Splits a large file (a time capture file or a waterfall file) over multiple disks
MMEMory:STORe:DTABle:TRACe[1   2   3   4]	command only	Saves the specified data table to a file on the specified disk
MMEM:STORe:LIMit:LOWer:TRACe[1   2   3   4]	command only	Saves the lower limit of the specified trace to a file on the specified disk
MMEM:STORe:LIMit:UPPer:TRACe[1   2   3   4]	command only	Saves the upper limit of the specified trace to a file on the specified disk
MMEMory:STORe:MATH	command only	Saves a complete set of math definitions to a file on the specified disk
MMEMory:STORe:PROGram	command only	Saves an Instrument BASIC program to a file on the specified disk
MMEMory:STORe:PROGram:FORMat	‡	Specifies the format Agilent Instrument BASIC programs are stored
MMEMory:STORe:STATe	command only	Saves the instrument state to a file on the specified disk
MMEMory:STORe:SYNTHeSis	command only	Stores a synthesis table to a file on the specified disk
MMEMory:STORe:TCAPtUre	command only	Saves the time capture buffer to a file on the specified disk
MMEMory:STORe:TRACe	command only	Saves the specified trace to a file on the specified disk
MMEMory:STORe:WATERfall	command only	Saves the current waterfall display to a file on the specified disk
<b>OUTPut</b>		
OUTPut:FILTer[:LPASs][:STATe]	‡	Turns on a low pass filter for the arbitrary source

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
OUTPut[:STATe]	‡	Enables the analyzer's internal source
<b>PROGram</b>		
PROGram:EDIT:ENABLE	‡	Disables the Agilent Instrument BASIC editor
PROGram:EXPLicit:DEFine	‡	Loads an Agilent Instrument BASIC program into the specified program buffer from an external controller
PROGram:EXPLicit:LABel	‡	Loads a softkey label for the specified
HP Instrument BASIC program		
PROGram[:SELEcted]:DEFine	‡	Loads an HP Instrument BASIC program from an external controller into the active program buffer
PROGram[:SELEcted]:DELete:ALL	command only	Deletes all HP Instrument BASIC programs stored in the analyzer
PROGram[:SELEcted]:DELete[:SELEcted]	command only	Deletes the active HP Instrument BASIC program
PROGram[:SELEcted]:LABel	‡	Loads a softkey label for the active HP Instrument BASIC program
PROGram[:SELEcted]:MALLocate	‡	Allocates memory space for HP Instrument BASIC programs
PROGram[:SELEcted]:NAME	‡	Selects an HP Instrument BASIC program
PROGram[:SELEcted]:NUMBER	‡	Loads a new value for the specified numeric variable in the active HP Instrument BASIC program
PROGram[:SELEcted]:STATe	‡	Selects the state of the active HP Instrument BASIC program
PROGram[:SELEcted]:STRing	‡	Loads a new value for the specified string variable for the active HP Instrument BASIC program
<b>[[:SENSE]]</b>		
[SENSE:]AVERage:CONFidence	‡	Specifies the confidence level used in equal confidence averaging in octave measurements
[SENSE:]AVERage:COUNT	‡	Specifies a count or a weighting factor for the averaged measurement data
[SENSE:]AVERage:HOLD	‡	Specifies the type of hold used in averaging octave measurements
[SENSE:]AVERage:IMPulse	‡	Enables impulse detection in octave measurements
[SENSE:]AVERage:IREsult:RATE	‡	Specifies how often the display is updated when fast average mode is on
[SENSE:]AVERage:IREsult[:STATe]	‡	Selects fast average mode
[SENSE:]AVERage:PREView	‡	Specifies the type of preview averaging
[SENSE:]AVERage:PREView:ACCept	command only	Accept the current time record during preview averaging
[SENSE:]AVERage:PREView:REJect	command only	Reject the current time record during preview averaging

‡ Command form; add "?" for query form



Subsystem Commands	Form	Description
[SENSe:]AVERage:PREView:TIME	‡	Specifies the amount of time the analyzer waits for a response in timed preview averaging
[SENSe:]AVERage[:STATe]	‡	Turns the selected averaging function (AVER:TYPE) on or off
[SENSe:]AVERage:TCONtrol	‡	Specifies how the analyzer behaves after the count (AVER:COUN) is reached
[SENSe:]AVERage:TIME	‡	Specify the time period used in averaging octave measurements and histograms
[SENSe:]AVERage:TYPE	‡	Specifies the type of averaging the analyzer performs
[SENSe:]FEED	‡	Specifies the data source for a measurement; either from the input channels or from the time capture buffer
[SENSe:]FREQuency:BLocksize	‡	Specifies the number of real-time data points displayed on the analyzer's screen
[SENSe:]FREQuency:CENTer	‡	Specifies the center frequency for the current measurement
[SENSe:]FREQuency:MANual	‡	Selects a discrete point to be measured during manual sweep mode
[SENSe:]FREQuency:RESolution	‡	Specifies the frequency measurement resolution for FFT and swept sine instrument modes
[SENSe:]FREQuency:RESolution:AUTO	‡	Selects auto resolution for swept sine instrument mode
[SENSe:]FREQuency:RESolution:AUTO:MCHange	‡	Specifies the maximum change permitted between the frequency response of the current measurement point and the frequency response of the previous measurement point
[SENSe:]FREQuency:RESolution:AUTO:MINimum	‡	Specifies the initial resolution of a swept sine measurement with automatic resolution
[SENSe:]FREQuency:RESolution:OCTave	‡	Specifies the type of octave measurement
[SENSe:]FREQuency:SPAN	‡	Specifies the frequency bandwidth to be measured
[SENSe:]FREQuency:SPAN:FULL	command only	Sets the analyzer to the widest frequency span available for the current instrument mode
[SENSe:]FREQuency:SPAN:LINK	‡	Specifies the frequency parameter which remains constant if frequency span or record length is modified
[SENSe:]FREQuency:START	‡	Specifies the start (lowest) frequency for the frequency band of the current measurement
[SENSe:]FREQuency:STEP[:INCRement]	‡	Specifies the step size which is used when changing frequency parameters
[SENSe:]FREQuency:STOP	‡	Sets the stop frequency to the specified value
[SENSe:]HISTogram:BINs	‡	Specifies the number of bins in a histogram
[SENSe:]ORDer:MAXimum	‡	Specifies the number of orders to be displayed
[SENSe:]ORDer:RESolution	‡	Specifies order resolution
[SENSe:]ORDer:RESolution:TRACk	‡	Specifies the number of points per order track
[SENSe:]ORDer:RPM:MAXimum	‡	Specifies the maximum rotational speed range you want to analyze

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
[SENSe:]ORDer:RPM:MINimum	‡	Specifies the minimum rotational speed range you want to analyze
[SENSe:]ORDer:TRACk[1   2   3   4   5]	‡	Specify the order number for the selected track
[SENSe:]ORDer:TRACk[1   2   3   4   5]:STATe	‡	Selects order track or order spectrum measurements
[SENSe:]REFerence	‡	Specifies the reference channel(s)
[SENSe:]REJect:STATe	‡	Turns overload rejection on or off
[SENSe:]SWEep:DIRectioN	‡	Specifies the direction of the sweep
[SENSe:]SWEep:DWELl	‡	Specifies the integration time for swept sine measurements
[SENSe:]SWEep:MODE	‡	Specifies automatic or manual sweep modes
[SENSe:]SWEep:OVERlap	‡	Specifies the maximum amount of time record overlap
[SENSe:]SWEep:SPACing	‡	Selects linear or logarithmic spacing between measurement data points
[SENSe:]SWEep:STIME	‡	Specifies the settling time for a swept sine measurement
[SENSe:]SWEep:TIME	‡	Specifies the length of the time record in seconds
[SENSe:]TCAPture:ABORt	command only	Stops the time capture process
[SENSe:]TCAPture:DELeTe	command only	Removes the time capture buffer
[SENSe:]TCAPture[:IMMEdiate]	command only	Starts the collection of data for the time capture process
[SENSe:]TCAPture:LENGth	‡	Specifies the length of the time capture buffer
[SENSe:]TCAPture:MALLocate	command only	Allocates memory for the time capture buffer
[SENSe:]TCAPture:STARt[1   2   3   4]	‡	Specifies the beginning of the time capture data used in a measurement
[SENSe:]TCAPture:STOP[1   2   3   4]	‡	Specifies the end of the time capture data used in a measurement
[SENSe:]TCAPture:TACHometer:RPM:MAXimum	‡	Specifies the tachometer's maximum RPM when included in the time capture buffer
[SENSe:]TCAPture:TACHometer[:STATe]	‡	Directs the analyzer to include the tachometer input signal in the time capture buffer
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:AUTO	‡	Automatically selects the best range on the specified channel for the current input signal
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:AUTO:DIRectioN	‡	Selects the type of autorange; up/down or up only
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:UNIT:USER:LABel	‡	Assigns a name to the transducer units for the specified input channel when VOLT:RANG:UNIT:XDCR:LABel is USER
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:UNIT:USER:SFACTor	‡	Specifies the transducer sensitivity for transducer units
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:UNIT:USER[:STATe]	‡	Enables the use of transducer units

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
[SENSe:]VOLTage[1   2   3   4][:DC]:RANGe:UNIT:XDCR:LABel	‡	Specifies a transducer unit for the specified channel
[SENS:]VOLT[1   2   3   4][:DC]:RANGe[:UPPer]	‡	Specifies the input range for the selected channel
[SENSe:]WINDow[1   2   3   4]:EXPOntial	‡	Specifies the time constant for the exponential window function
[SENSe:]WINDow[1   2   3   4]:FORCe	‡	Specifies the width of the force window
[SENSe:]WINDow[1   2   3   4]:ORDer:DC	‡	Directs the analyzer to include the DC bin in the composite power calculation (order track measurements)
[SENSe:]WINDow[1   2   3   4][:TYPE]	‡	Selects the type of windowing function
<b>SOURce</b>		
SOURce:BURSt	‡	Sets the burst length for the burst source types
SOURce:FREQuency[:CW]	‡	Sets the frequency of the sine source
SOURce:FREQuency:FIXed	‡	Sets the frequency of the sine source type
SOURce:FUNcTion[:SHApe]	‡	Specifies the source output
SOURce:USER:CAPTure	‡	Specifies which channel of time capture data to use
SOURce:USER[:REGister]	‡	Specifies the data register which contains the data for the arbitrary source
SOURce:USER:REPeat	‡	Enables the source repeat function
SOURce:VOLTage[:LEVel]:AUTO	‡	Enables the autolevel feature in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]	‡	Specifies the source output level
SOURce:VOLTage[:LEVel][:IMMEDIATE]:OFFSet	‡	Specifies a DC offset for the source output
SOURce:VOLTage[:LEVel]:REFerence	‡	Specifies the amplitude of the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage[:LEVel]:REFerence:CHANnel	‡	Selects the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage[:LEVel]:REFerence:TOLerance	‡	Specifies the sensitivity of the autolevel algorithm in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage:LIMit[:AMPLitude]	‡	Sets the maximum limit used by the autolevel algorithm to adjust the source's amplitude in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage:LIMit:INPut	‡	Sets the maximum amplitude of the <i>non-reference</i> input channels for the autolevel feature in swept sine measurements (INST:SEL SINE)
SOURce:VOLTage:SLEW	‡	Specifies the source amplitude ramp rate in swept sine measurements (INST:SEL SINE)
<b>STATus</b>		

‡ Command form; add "?" for query form

Agilent 35670A Command Summary

Subsystem Commands	Form	Description
STATus:DEvice:CONDition?	query only	Reads and clears the Device State condition register
STATus:DEvice:ENABle	‡	Sets and queries bits in the Device State enable register
STATus:DEvice[:EVENT]?	query only	Reads and clears the Device State event register
STATus:DEvice:NTRansition	‡	Sets and queries bits in the Device Status negative transition register
STATus:DEvice:PTRansition	‡	Sets and queries bits in the Device State positive transition register
STATus:OPERation:CONDition?	query only	Reads the Operation Status condition register
STATus:OPERation:ENABle	‡	Sets and queries bits in the Operation Status enable register
STATus:OPERation[:EVENT]?	query only	Reads and clears the Operation Status event register
STATus:OPERation:NTRansition	‡	Sets and queries bits in the Operation Status negative transition register
STATus:OPERation:PTRansition	‡	Sets bits in the Operation Status positive transition register
STATus:PRESet	command only	Sets bits in most enable and transition registers to their default state
STATus:QUEStionable:CONDition?	query only	Reads and clears the Questionable Status condition register
STATus:QUEStionable:ENABle	‡	Sets and queries bits in the Questionable Status enable register
STATus:QUEStionable[:EVENT]?	query only	Reads and clears the Questionable Status event register
STATus:QUEStionable:LIMit:CONDition?	query only	Reads and clears the Limit Fail condition register
STATus:QUEStionable:LIMit:ENABle	‡	Sets and queries bits in the Limit Fail enable register
STATus:QUEStionable:LIMit[:EVENT]?	query only	Reads and clears the Limit Fail event register
STATus:QUEStionable:LIMit:NTRansition	‡	Sets and queries bits in the Limit Fail negative transition register
STATus:QUEStionable:LIMit:PTRansition	‡	Sets queries bits in the Limit Fail positive transition register
STATus:QUEStionable:NTRansition	‡	Sets and queries bits in the Questionable Status negative transition register
STATus:QUEStionable:PTRansition	‡	Sets and queries bits in the Questionable Status positive transition register
STATus:QUEStionable:VOLTage:CONDition?	query only	Reads the Questionable Voltage condition register
STATus:QUEStionable:VOLTage:ENABle	‡	Sets and queries bits in the Questionable Voltage enable register
STATus:QUEStionable:VOLTage[:EVENT]?	query only	Reads and clears the Questionable Voltage event register
STATus:QUEStionable:VOLTage:NTRansition	‡	Sets and queries bits in the Questionable Voltage negative transition register
STATus:QUEStionable:VOLTage:PTRansition	‡	Sets bits in the Questionable Voltage positive transition register
STATus:USER:ENABle	‡	Sets and queries bits in the User Defined enable register
STATus:USER[:EVENT]?	query only	Reads and clears the User Defined event register

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
STATus:USER:PULSe	command only	Sets bits in the User Defined event register
<b>SYSTEM</b>		
SYSTem:BEEPer[:IMMediate]	command only	Generates the tone at a given frequency from the analyzer's beeper
SYSTem:BEEPer:STATe	‡	Enables the analyzer's beeper
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	‡	Sets the analyzer's GPIB address
SYSTem:COMMunicate:SERial[:RECeive]:BAUD	‡	Specifies the data-transfer rate between the analyzer and RS-232-C peripheral devices
SYSTem:COMMunicate:SERial[:RECeive]:BITS	‡	Specifies the number of bits in a character for the RS-232-C interface
SYSTem:COMMunicate:SERial[:RECeive]:PACE	‡	Sets the RS-232-C receiver handshake pacing type
SYSTem:COMMunicate:SERial[:RECeive]:PARity:CHECK	‡	Enables RS-232-C parity verification capability
SYSTem:COMMunicate:SERial[:RECeive]:PARity[:TYPE]	‡	Sets the parity generated for characters transmitted over the RS-232-C interface
SYSTem:COMMunicate:SERial[:RECeive]:SBITs	‡	Specifies the number of "stop bits" sent with each character over the RS-232-C interface
SYSTem:COMMunicate:SERial:TRANsmit:PACE	‡	Sets the RS-232 transmitter handshake pacing type
SYSTem:DATE	‡	Sets the date in the analyzer's battery-backed clock
SYSTem:ERRor?	query only	Returns one error message from the analyzer's error queue
SYSTem:FAN[:STATe]	‡	Sets the fan's operating mode
SYSTem:FLOG:CLEar	command only	Clears the fault log of all entries
SYSTem:KEY	‡	Writes or queries front-panel key presses
SYSTem:KLOCK	‡	Disables the keyboard
SYSTem:POWer:SOURce?	query only	Queries the setting of the rear-panel power select switch
SYSTem:POWer:STATe	‡	Turns off the analyzer's power
SYSTem:PRESet	command only	Returns most of the analyzer's parameters to their preset states
SYSTem:SET	‡	Transfers an instrument state between the analyzer and an external controller
SYSTem:TIME	‡	Sets the time in the analyzer's battery-backed clock
SYSTem:VERSion?	query only	Returns the SCPI version to which the analyzer complies
<b>TEST</b>		
TEST:LOG:CLEar	command only	Clears the test log
TEST:LONG	command only	Executes the long confidence test
TEST:LONG:RESult?	query only	Returns the overall result of the long confidence test

‡ Command form; add "?" for query form

Subsystem Commands	Form	Description
<b>TRACe</b>		
TRACe[:DATA]	‡	Stores data to the specified data register
TRACe:WATERfall[:DATA]	‡	Stores data to the specified waterfall register
TRACe:X[:DATA]?	query only	Returns the X-axis data for trace displays
TRACe:X:UNIT?	query only	Returns the unit for the x-axis for trace displays
TRACe:Z[:DATA]?	query only	Returns the Z-axis data for waterfall displays
TRACe:Z:UNIT?	query only	Returns the unit for the z-axis in waterfall displays
<b>TRIGger</b>		
TRIGger:EXTernal:FILTer[:LPAS][:STATe]	‡	Turns on a low pass filter for the external trigger
TRIGger:EXTernal:LEVel	‡	Specifies the level of the external input signal which causes the analyzer to trigger
TRIGger:EXTernal:RANGe	‡	Selects the range for the external trigger's level
TRIGger[:IMMediate]	command only	Triggers the analyzer if TRIG:SOUR is BUS
TRIGger:LEVel	‡	Specifies the level of the input signal which causes the analyzer to trigger
TRIGger:SLOPe	‡	Specifies the slope of the signal which triggers the analyzer
TRIGger:SOURce	‡	Selects the source of the trigger event
TRIGger:STARt[1   2   3   4]	‡	Specifies a pre-trigger or a post-trigger delay for the specified channel
TRIGger:TACHometer:HOLDoff	‡	Specifies a time interval in which the tachometer trigger is inhibited
TRIGger:TACHometer:LEVel	‡	Specifies the level of the tachometer's input signal which causes the analyzer to trigger
TRIGger:TACHometer:PCOunt	‡	Specifies the number of tachometer pulses that occur in one revolution of the shaft
TRIGger:TACHometer:RANGe	‡	Specifies the input range of the analyzer's tachometer
TRIGger:TACHometer[:RPM]?	query only	Reads and returns the current RPM value for the analyzer's tachometer
TRIGger:TACHometer:SLOPe	‡	Specifies the slope of the tachometer input signal to be used in RPM step arming or order tracking

‡ Command form; add "?" for query form

‡ Command form; add "?" for query form

## **Cross-Reference from Front-Panel Keys to GPIB Commands**

---





## Cross-Reference from Front-Panel Keys to GPIB Commands

---

### Introduction

This section lists analyzer hardkeys and softkeys and their equivalent GPIB commands.

Softkeys are indented to indicate their position in the menu tree. Keys that do not have an equivalent GPIB command are excluded from the list. Keys which appear in multiple menus may appear only once. For example, time capture is available in most instrument modes and appears in multiple Measurement Data hardkey menus. The GPIB command for time capture is only listed once.

Multiple GPIB commands may be required for a single softkey. In these cases, the entire command string is listed. The command string includes valid parameter values.

You can also determine the equivalent GPIB command for any key sequence by turning on the GPIB echo facility. With GPIB echo on, the analyzer displays the equivalent GPIB command for each front panel key sequence. To turn on GPIB echo, press the following keys:

```
[ Local/GPIB ]  
  [ GPIB ECHO ON OFF ] to highlight ON.
```

## Measurement Group

### Front Panel Key

### GPIB Command

Inst Mode
-----------

[ FFT ANALYSIS ]	INSTrument:SElect FFT
[ OCTAVE ANALYSIS ]	INSTrument:SElect OCTave
[ ORDER ANALYSIS ]	INSTrument:SElect ORDer
[ SWEPT SINE ]	INSTrument:SElect SINE
[ CORRELATN ANALYSIS ]	INSTrument:SElect CORRelation
[ HISTOGRAM / TIME ]	INSTrument:SElect HISTogram
[ CAPTURE ON OFF ]	[SENSe:]FEED INPut   TCAPture
[ 1 CHANNEL ]	INPut2[:STATe] OFF
[ 2 CHANNELS ]	INPut2[:STATe] ON
[ 4 CHANNELS ]	INPut4[:STATe] ON
[ REF CHANNELS 1 1,3 ]	[SENSe:]REFerence SINGle   PAIR
[ TIME CAPTURE ]	
[ MEAS FROM INP BUFR ]	[SENSe:]FEED INPut   TCAPture
[ FILL BUFFER ]	[SENSe:]TCAPture[:IMMediate]
[ ABORT FILL ]	[SENSe:]TCAPture:ABORt
[ BUFFER LENGTH ]	[SENSe:]TCAPture:LENGth 0 ~ 9.9e+37[S   BLK   PNT]
[ ALLOCATE BUFFER ]	
[ CONFIRM ALLOCATE ]	[SENSe:]TCAPture:MALLocate
[ REMOVE CAPTURE ]	
[ CONFIRM REMOVE ]	[SENSe:]TCAPture:DELeTe
[ TACHOMETR OPTIONS ]	
[ TACH DATA ON OFF ]	[SENSe:]TCAPture:TACHometer[:STATe] OFF   0   ON   1
[ MAX RPM ]	[SENSe:]TCAPture:TACHometer:RPM:MAXimum 0 ~ 9.9e+37
[ ANALYSIS REGION ]	
[ STRT TIME CHANNEL 1 ]	[SENSe:]TCAPture:STARt1 0 ~ 9.9e+37[S   BLK   PNT]
[ STOP TIME CHANNEL 1 ]	[SENSe:]TCAPture:STOP1 0 ~ 9.9e+37[S   BLK   PNT]
[ STRT TIME CHANNEL 2 ]	[SENSe:]TCAPture:STARt2 0 ~ 9.9e+37[S   BLK   PNT]
[ STOP TIME CHANNEL 2 ]	[SENSe:]TCAPture:STOP2 0 ~ 9.9e+37[S   BLK   PNT]
[ STRT TIME CHANNEL 3 ]	[SENSe:]TCAPture:STARt3 0 ~ 9.9e+37[S   BLK   PNT]
[ STOP TIME CHANNEL 3 ]	[SENSe:]TCAPture:STOP3 0 ~ 9.9e+37[S   BLK   PNT]
[ STRT TIME CHANNEL 4 ]	[SENSe:]TCAPture:STARt4 0 ~ 9.9e+37[S   BLK   PNT]
[ STOP TIME CHANNEL 4 ]	[SENSe:]TCAPture:STOP4 0 ~ 9.9e+37[S   BLK   PNT]
[ ENVELOPE ON OFF ]	DISPlay:TCAPture:ENVELOpe ON OFF

Front Panel Key

GPIB Command

**Freq**

[ SPAN ]	[SENSe:]FREQuency:SPAN 0.015625 ~ 102400[HZ]
[ CENTER ]	[SENSe:]FREQuency:CENTer 0.0234375 ~ 115000[HZ]
[ START ]	[SENSe:]FREQuency:STARt 0 ~ 115000[HZ]
[ STOP ]	[SENSe:]FREQuency:STOP 0.03125 ~ 115000[HZ]
[ ZERO START ]	[SENSe:]FREQuency:STARt 0
[ FULL SPAN ]	[SENSe:]FREQuency:SPAN:FULL
[ ENTRY STEP SIZE ]	[SENSe:]FREQuency:STEP[:INCRement] 0.015625 ~ 10240[HZ]
[ RECORD LENGTH ]	[SENSe:]SWEep:TIME 0.000976562 ~ 8192[S]
[ RESOLUTN LINES ]	[SENSe:][SENSe:]FREQuency:RESolution 100   200   400   800
[ 1/1 OCTAVE ]	[SENSe:]FREQuency:RESolution:OCTave FULL
[ 1/3 OCTAVE ]	[SENSe:]FREQuency:RESolution:OCTave THIRd
[ 1/12 OCTAVE ]	[SENSe:]FREQuency:RESolution:OCTave TWELth
[ MIN RPM ]	[SENSe:]ORDer:RPM:MINimum 0 ~ 9.9e+37
[ MAX RPM ]	[SENSe:]ORDer:RPM:MAXimum 0 ~ 9.9e+37
[ MAX ORDER ]	[SENSe:]ORDer:MAXimum 0 ~ 9.9e+37[ORD]
[ DELTA ORDER ]	[SENSe:]ORDer:RESolution 0.0078125 ~ 1[ORD]
[ TRACK ON OFF ]	[SENSe:]ORDer:TRACk:STATe OFF   0   ON   1
[ TRACK SETUP ]	
[ TRACK 1 ORDER ]	[SENSe:]ORDer:TRACk1 0 ~ 9.9e+37[ORD]
[ TRACK 2 ORDER ]	[SENSe:]ORDer:TRACk2 0 ~ 9.9e+37[ORD]
[ TRACK 3 ORDER ]	[SENSe:]ORDer:TRACk3 0 ~ 9.9e+37[ORD]
[ TRACK 4 ORDER ]	[SENSe:]ORDer:TRACk4 0 ~ 9.9e+37[ORD]
[ TRACK 5 ORDER ]	[SENSe:]ORDer:TRACk5 0 ~ 9.9e+37[ORD]
[ TRACK POINTS ]	[SENSe:]ORDer:RESolution:TRACk 1 ~ 2048
[ SWEEP UP DOWN ]	[SENSe:]SWEep:DIRection UP   DOWN
[ SWEEP AUTO MAN ]	[SENSe:]SWEep:MODE AUTO   MANUal
[ MANUAL FREQ ]	[SENSe:]FREQuency:MANUal 0.015625 ~ 51200[HZ]
[ RESOLUTN SETUP ]	
[ RESOLUTN ]	[SENSe:]FREQuency:RESolution 0.015625 ~ 51200 [HZ   PCT   PNT/SWP   PNT/DEC   PNT/OCT]
[ SWEEP LIN LOG ]	[SENSe:]SWEep:SPACing LINear   LOGarithmic
[ AUTO RES ON OFF ]	[SENSe:]FREQuency:RESolution:AUTO OFF   0   ON   1
[ MAXIMUM % CHANGE ]	[SENSe:]FREQuency:RESolution:AUTO:MCHange 0.391 ~ 100
[ MINIMUM RESOLUTN ]	[SENSe:]FREQuency:RESolution:AUTO:MINimum 0.015625 ~ 51200 [HZ   PCT   PNT/SWP   PNT/DEC   PNT/OCT]

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

[ SAMPLE TIME ]	[SENSe:]SWEEp:TIME 0.000976562 ~ 8192[S]
[ RECORD TIME ]	[SENSe:]SWEEp:TIME 0.000976562 ~ 8192[S]
[ BLOCKSIZE ]	[SENSe:]FREQuency:BLOCkszie 256 ~ 2048
[ HISTOGRAM LENGTH ]	[SENSe:]AVERAge:TIME 0 ~ 9.9e+ 37[S   REC   PNT]
[ HISTOGRAM BINS ]	[SENSe:]HISTOgram:BINS 4 ~ 1024
[ RECORD LENGTH ]	[SENSe:]SWEEp:TIME 0.000976562 ~ 8192[S]

Window

[ HANNING ]	[SENSe:]WINDow[:TYPE] HANNing
[ FLAT TOP ]	[SENSe:]WINDow[:TYPE] FLATtop
[ UNIFORM ]	[SENSe:]WINDow[:TYPE] UNIFORM
[ FORCE EXPO ]	[SENSe:]WINDow1[:TYPE] FORCe;WINDow2[:TYPE] EXP;WINDow3[:TYPE] EXP; WINDow4[:TYPE] EXP
[ CHANNEL 1 FORC EXPO ]	[SENSe:]WINDow1[:TYPE] FORCe   EXP
[ CHANNEL 2 FORC EXPO ]	[SENSe:]WINDow2[:TYPE] FORCe   EXP
[ CHANNEL 3 FORC EXPO ]	[SENSe:]WINDow3[:TYPE] FORCe   EXP
[ CHANNEL 4 FORC EXPO ]	[SENSe:]WINDow4[:TYPE] FORCe   EXP
[ FORCE WIDTH ]	[SENSe:]WINDow:FORCe 3.8147e-06 ~ 9999900[S]
[ EXPO DECAY ]	[SENSe:]WINDow:EXPonential 3.8147e-06 ~ 9999900[S]
[ CP DC BIN ON OFF ]	[SENSe:]WINDow:ORDer:DC OFF   0   ON   1
[ ZERO PAD -T/4, T/4 ]	[SENSe:]WINDow[:TYPE] LLAG
[ ZERO PAD 0, T/2 ]	[SENSe:]WINDow[:TYPE] LAG

Input

[ CHANNEL 1   2   3   4 RANGE ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe[:UPPer] -51 ~ 31.66 [DBVRMS   VPK   DBVPK   V   DBV   EU   DBEU   VRMS]
[ CH FIXED RANGE ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:AUTO OFF
[ CH AUTO UP ONLY ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:AUTO ON;AUTO:DIRrection UP
[ CH AUTO RANGE ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:AUTO ON;AUTO:DIRrection EITHER
[ FRONT END CH1   2   3   4 SETUP ]	
[ INPUT LOW FLOAT GND ]	INPut1   2   3   4:LOW GROund   FLOat
[ COUPLING AC DC ]	INPut1   2   3   4:COUPling AC   DC
[ ANTIALIAS ON OFF ]	INPut1   2   3   4:FILTer[:LPASs][:STATe] OFF   0   ON   1
[ A WT FLTR ON OFF ]	INPut1   2   3   4:FILTer:AWEighting[:STATe] OFF   0   ON   1
[ ICP SUPPLY ON OFF ]	INPut1   2   3   4:BIAS[:STATe] OFF   0   ON   1

Front Panel Key	GPIB Command
[ XDCR UNIT CH1   2   3   4 SETUP ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER [:STATe] OFF   0   ON   1
[ XDCR UNIT ON OFF ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:SFACTOR
[ XDCR UNIT SENSITIVITY ]	-9.9e+37 ~ 9.9e+37[V/EU   EU/V]
[ XDCR UNIT LABEL ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'PA'
[ Pascal ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'G'
[ g ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'M/S2'
[ m/s^2 ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'M/S'
[ m/s ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'M'
[ m ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'KG'
[ kg ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'N'
[ N ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'DYN'
[ dyn ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'INCH/S2'
[ inch/s^2 ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'INCH/S'
[ inch/s ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'INCH'
[ inch ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel 'MIL'
[ mil ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel ':LB'
[ lb ]	[SENSe:]VOLTage1   2   3   4[:DC]:RANGe:UNIT:USER:LABel '< STRING >'
[ USER UNIT LABEL ]	
[ TACHOMETR SETUP ]	
[ TACH PULS PER REV ]	TRIGger:TACHometer:PCOUNT 0.5 ~ 2048
[ TRG RANGE +/- 20 4 ]	TRIGger:TACHometer:RANGe HIGH   LOW
[ LEVEL ]	TRIGger:TACHometer:LEVel -20 ~ 20[V]
[ HOLDOFF TIME ]	TRIGger:TACHometer:HOLDoff 0 ~ 0.052224[S]
[ SLOPE POS NEG ]	TRIGger:TACHometer:SLOPe POSitive   NEGative
[ TACH DISP ON OFF ]	DISPlay:RPM OFF   0   ON   1
<b>Source</b>	
[ SOURCE ON OFF ]	OUTPut[:STATe] OFF   0   ON   1
[ LEVEL ]	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude] (-9.9e+37 ~ 13.9794[DBVRMS   VPK   DBVPK   V   DBV   VRMS])
[ DC OFFSET ]	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]:OFFSET (-9.9e+37 ~ 13.9794[DBVRMS   VPK   DBVPK   V   DBV   VRMS])
[ RANDOM NOISE ]	SOURce:FUNCTion[:SHAPE] RANDom
[ PERIODIC CHIRP ]	SOURce:FUNCTion[:SHAPE] PCHirp
[ PINK NOISE ]	SOURce:FUNCTion[:SHAPE] PINK
[ FIXED SINE ]	SOURce:FUNCTion[:SHAPE] SIN;FREQuency[:CW] 0 ~ 115000[HZ]
[ BURST RANDOM ]	SOURce:FUNCTion[:SHAPE] BRAN;BURSt 0 ~ 100[PCT]
[ BURST CHIRP ]	SOURce:FUNCTion[:SHAPE] BRAN;BURSt 0 ~ 100[PCT]
[ ARBITRARY D1-D8 ]	SOURce:FUNCTion[:SHAPE] USER [D1   D2   D3   D4   D5   D6   D7   D8]
[ CAPTURE CHANNEL ]	SOURce:USER:CAPTure 1 ~ 4

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key	GPIB Command
[ ARB SRC SETUP ]	
[ REPEAT ON OFF ]	SOURce:USER:REPeat OFF   0   ON   1
[ DATA REG D1 ]	SOURce:USER[:REGister] D1
[ DATA REG D2 ]	SOURce:USER[:REGister] D2
[ DATA REG D3 ]	SOURce:USER[:REGister] D3
[ DATA REG D4 ]	SOURce:USER[:REGister] D4
[ DATA REG D5 ]	SOURce:USER[:REGister] D5
[ DATA REG D6 ]	SOURce:USER[:REGister] D6
[ DATA REG D7 ]	SOURce:USER[:REGister] D7
[ DATA REG D8 ]	SOURce:USER[:REGister] D8
[ ARB FILTER ON OFF ]	OUTPut:FILTer OFF   0   ON   1
[ RAMP RATE ]	SOURce:VOLTage:SLEW 0 ~ 10000 [V/S   VPK/S   VRMS/S]
[ AUTOLEVEL ON OFF ]	SOURce:VOLTage[:LEVel]:AUTO OFF   0   ON   1
[ AUTOLEVEL SETUP ]	
[ REF CHAN CH1 CH2 CH3 CH4 ]	SOURce:VOLTage[:LEVel]:REFerence:CHANnel INPut1   INPut2   INPut3   INPut4
[ REFERENCE LEVEL ]	SOURce:VOLTage[:LEVel]:REFerence -69.276 ~ 31.66 [DBVRMS   VPK   DBVPK   V   DBV   EU   DBEU   VRMS]
[ REFERENCE TOLERANCE ]	SOURce:VOLTage[:LEVel]:REFerence:TOLerance 0.1 ~ 20[DB]
[ MAX SRC LEVEL ]	SOURce:VOLTage:LIMit[:AMPLitude] -9.9e+37 ~ 13.9794 [DBVRMS   VPK   DBVPK   V   DBV   VRMS]
[ MAX INPUT LEVEL ]	SOURce:VOLTage:LIMit:INPut -69.276 ~ 31.66 [DBVRMS   VPK   DBVPK   V   DBV   EU   DBEU   VRMS]
<b>Trigger</b>	
[ FREE RUN TRIGGER ]	TRIGger:SOURce IMMEDIATE
[ EXTERNAL TRIGGER ]	TRIGger:SOURce EXTernal
[ CHANNEL 1   2   3   4 ]	TRIGger:SOURce INTernal1   2   3   4
[ SOURCE TRIGGER ]	TRIGger:SOURce OUTPut
[ GPIB TRIGGER ]	TRIGger:SOURce BUS
[ TACHOMETR SETUP ]	
[ TACH PULS PER REV ]	TRIGger:TACHometer:PCOunt 0.5 ~ 2048
[ TRG RANGE +/- 20 4 ]	TRIGger:TACHometer:RANGe HIGH   LOW
[ LEVEL ]	TRIGger:TACHometer:LEVel -20 ~ 20[V]
[ HOLDOFF TIME ]	TRIGger:TACHometer:HOLDoff 0 ~ 0.052224[S]
[ SLOPE POS NEG ]	TRIGger:TACHometer:SLOPe POSitive   NEGative
[ TRIGGER SETUP ]	
[ CHANNEL LEVEL ]	TRIGger:LEVel -9.9e+37 ~ 9.9e+37 [PCT   V   VPK   EU]
[ EXT LEVEL USER ]	TRIGger:EXTernal:RANGe HIGH   LOW;LEVel -9.9e+37 ~ 9.9e+37 [PCT   V   VPK   EU]
[ EXT LEVEL TTL ]	TRIGger:EXTernal:RANGe HIGH;LEVel 1.484375
[ USER EXT LEVEL ]	TRIGger:EXTernal:LEVel -10 ~ 10[V]
[ EXT RANGE +/- 10 2 ]	TRIGger:EXTernal:RANGe HIGH   LOW
[ SLOPE POS NEG ]	TRIGger:SLOPe POSitive   NEGative

[ EXT FILTER ON OFF ]	TRIGger:EXtErnal:FiLTeR OFF   0   ON   1
[ CHANNEL 1   2   3   4 DELAY ]	TRIGger:STARt1   2   3   4 -9.9e+37 ~ 9.9e+37[S]
[ ARM SETUP ]	
[ AUTOMATIC ARM ]	ARM:SOURce IMMEDIATE
[ MANUAL ARM ]	ARM:SOURce MANUal
[ RPM STEP ARM ]	ARM:SOURce RPM
[ TIME STEP ARM ]	ARM:SOURce TiMEr
[ START RPM USAGE ]	
[ START RPM OFF ]	ARM:RPM:MODE OFF
[ RPM INCREASNG ]	ARM:RPM:MODE UP
[ RPM DECREASNG ]	ARM:RPM:MODE DOWN
[ START RPM ]	ARM:RPM:THReshold 5 ~ 491520
[ RPM STEP SIZE ]	ARM:RPM:INCRement 1 ~ 500000
[ TIME STEP SIZE ]	ARM:TiMEr 0 ~ 500000[S]
[ WATERFALL STEPS ]	CALCulate:WATerfall:COUNt 1 ~ 32760
[ ARM ]	ARM[:IMMEDIATE]
<b>Start</b>	
START Hardkey	ABORt;:INITiate[:IMMEDIATE]
<b>PauseCont</b>	
PAUSE/CONT Hardkey	INITiate:CONTinuous OFF   0   ON   1
<b>Avg</b>	
[ AVERAGE ON OFF ]	[SENSe:]AVERAge[:STATe] OFF   0   ON   1
[ NUMBER AVERAGES ]	[SENSe:]AVERAge:COUNt 1 ~ 1e+07
[ AVERAGE TYPE ]	
[ RMS ]	[SENSe:]AVERAge:TYPE RMS
[ RMS EXPONENTL ]	[SENSe:]AVERAge:TYPE RMS;TCONtrol EXPonential
[ TIME ]	[SENSe:]AVERAge:TYPE TIME
[ TIME EXPONENTL ]	[SENSe:]AVERAge:TYPE TIME;TCONtrol EXPonential
[ PEAK HOLD ]	[SENSe:]AVERAge:TYPE MAXimum
[ FAST AVG ON OFF ]	[SENSe:]AVERAge:IRESult[:STATe] OFF   0   ON   1
[ UPDATE RATE ]	[SENSe:]AVERAge:IRESult:RATE 1 ~ 99999
[ REPEAT ON OFF ]	[SENSe:]AVERAge:TCONtrol FREeze   REPeat   EXPonential
[ OVERLAP PERCENT ]	[SENSe:]SWEep:OVERlap 0 ~ 99[PCT]
[ OVLD REJ ON OFF ]	[SENSe:]REJect:STATe OFF   0   ON   1

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

[ AVERAGE PREVIEW ]	
[ PREVIEW OFF ]	[SENSe:]AVERAge:PREView OFF
[ MANUAL PREVIEW ]	[SENSe:]AVERAge:PREView MANUal
[ TIMED PREVIEW ]	[SENSe:]AVERAge:PREView TIMed
[ TIME ]	[SENSe:]AVERAge:PREView:TIME 0.1 ~ 3600[S]
[ REJECT TIME REC ]	[SENSe:]AVERAge:PREView:REJect
[ ACCEPT TIME REC ]	[SENSe:]AVERAge:PREView:ACCept
[ LINEAR ]	[SENSe:]AVERAge:TYPE RMS
[ EXPONENTL ]	[SENSe:]AVERAge:TYPE RMS;TCONtrol EXPonential
[ EQUAL CONFID ]	[SENSe:]AVERAge:TYPE ECONfidence
[ PEAK HOLD ]	[SENSe:]AVERAge:TYPE MAX
[ HOLD SETUP ]	
[ OFF ]	[SENSe:]AVERAge:HOLD OFF
[ MAXIMUM ]	[SENSe:]AVERAge:HOLD MAXimum
[ MINIMUM ]	[SENSe:]AVERAge:HOLD MINimum
[ AVERAGE TIME ]	[SENSe:]AVERAge:TIME 0.125 ~ 8192[S]
[ CONFIDNCE LEVEL ]	[SENSe:]AVERAge:CONFidence 0 ~ 100[DB]
[ IMPULSE ON OFF ]	[SENSe:]AVERAge:IMPulse OFF   0   ON   1
[ SETTLE TIME ]	[SENSe:]SWEep:STIME 0 ~ 9.9e+37[S   CYCLE]
[ INTEGRATE TIME ]	[SENSe:]SWEep:DWELI 0.00025 ~ 32768[S   CYCLE]



## Display Group

### Front Panel Key

### GPIB Command

Meas Data
-----------

[ PWR SPEC CHANNEL 1 2 3 4 ]	CALCulate#:FEED 'XFR:POW 1   2   3   4';MATH:STATe OFF;*WAI
[ LIN SPEC CHANNEL 1   2   3   4 ]	CALCulate#:FEED 'XFR:POW:LIN 1   2   3   4';MATH:STATe OFF;*WAI
[ TIME CHANNEL 1   2   3   4 ]	CALCulate#:FEED 'XTIM:VOLT 1   2   3   4';MATH:STATe OFF;*WAI
[ FREQ RESP 2/1   3/1   4/1   4/3 ]	CALCulate#:FEED 'XFR:POW:RATio 2,1   3,1   4,1   4,3';MATH:STATe OFF;*WAI
[ COHERENCE 2/1   3/1   4/1   4/3 ]	CALCulate#:FEED 'XFR:POW:COH 1,2   1,3   1,4   3,4';MATH:STATe OFF;*WAI
[ CROS SPEC 2/1   3/1   4/1   4/3 ]	CALCulate :FEED 'XFR:POW:CROS 1,2   1,3   1,4   3,4';MATH:STATe OFF;*WAI
[ MORE ]	
[ ORBIT 2/1   3/1   4/1   4/3 ]	CALCulate :FEED 'XVOL:VOLT 1,2   3,1   4,1   4,3';MATH:STATe OFF;*WAI
[ WINDOWED TIME CH1   2   3   4 ]	CALCulate#:FEED 'XTIM:VOLT:WINDow 1   2   3   4';MATH:STATe OFF;*WAI
[ CAPTURE CHANNEL 1   2   3   4 ]	CALCulate#:FEED 'TCAP 1   2   3   4';MATH:STATe OFF;*WAI
[ MATH FUNCTION ]	
[ F1 ]	CALCulate#:MATH:SElect F1;STATe ON
[ F2 ]	CALCulate#:MATH:SElect F2;STATe ON
[ F3 ]	CALCulate#:MATH:SElect F3;STATe ON
[ F4 ]	CALCulate#:MATH:SElect F4;STATe ON
[ F5 ]	CALCulate#:MATH:SElect F5;STATe ON
[ DATA REGISTER ]	
[ D1 ]	CALCulate#:FEED 'D1';MATH:STATe OFF;*WAI
[ D2 ]	CALCulate#:FEED 'D2';MATH:STATe OFF;*WAI
[ D3 ]	CALCulate#:FEED 'D3';MATH:STATe OFF;*WAI
[ D4 ]	CALCulate#:FEED 'D4';MATH:STATe OFF;*WAI
[ D5 ]	CALCulate#:FEED 'D5';MATH:STATe OFF;*WAI
[ D6 ]	CALCulate#:FEED 'D6';MATH:STATe OFF;*WAI
[ D7 ]	CALCulate#:FEED 'D7';MATH:STATe OFF;*WAI
[ D8 ]	CALCulate#:FEED 'D8';MATH:STATe OFF;*WAI
[ WATERFALL REGISTER ]	
[ W1 ]	CALCulate#:FEED 'W1';MATH:STATe OFF;*WAI
[ W2 ]	CALCulate#:FEED 'W2';MATH:STATe OFF;*WAI
[ W3 ]	CALCulate#:FEED 'W3';MATH:STATe OFF;*WAI
[ W4 ]	CALCulate#:FEED 'W4';MATH:STATe OFF;*WAI
[ W5 ]	CALCulate#:FEED 'W5';MATH:STATe OFF;*WAI
[ W6 ]	CALCulate#:FEED 'W6';MATH:STATe OFF;*WAI
[ W7 ]	CALCulate#:FEED 'W7';MATH:STATe OFF;*WAI
[ W8 ]	CALCulate#:FEED 'W8' ;MATH:STATe OFF;*WAI

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

[ MORE CHOICES ]

[ COMP PWR CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XFR:POW:COMP 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

[ ORDER TRK CHANNEL 1 2 3 4 ]

[ TRACK 1 ]

CALCulate#:FEED 'XORDer:TRACk 1 | 2 | 3 | 4,1';MATH:STATe OFF;\*WAI

[ TRACK 2 ]

CALCulate#:FEED 'XORDer:TRACk 1 | 2 | 3 | 4,2';MATH:STATe OFF;\*WAI

[ TRACK 3 ]

CALCulate#:FEED 'XORDer:TRACk 1 | 2 | 3 | 4,3';MATH:STATe OFF;\*WAI

[ TRACK 4 ]

CALCulate#:FEED 'XORDer:TRACk 1 | 2 | 3 | 4,4';MATH:STATe OFF;\*WAI

[ TRACK 5 ]

CALCulate#:FEED 'XORDer:TRACk 1 | 2 | 3 | 4,5';MATH:STATe OFF;\*WAI

[ RPM PROFILE ]

CALCulate#:FEED 'XRPM:PROF';MATH:STATe OFF;\*WAI

[ NORM VAR CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XFR:POW:VAR 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

[ AUTO CORR CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XTIM:CORR 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

[ CROS CORR 2/1 3/1 4/1 4/3 ]

CALCulate#:FEED 'XTIM:CORR:CROS 1,2 | 1,3 | 1,4 | 3,4';MATH:STATe OFF;\*WAI

[ HISTOGRAM CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XTIM:VOLT:HIST | 2 | 3 | 41';MATH:STATe OFF;\*WAI

[ PDF CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XTIM:VOLT:PDF 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

[ CDF CHANNEL 1 2 3 4 ]

CALCulate#:FEED 'XTIM:VOLT:CDF 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

[ UNFILTERD TIME CH 1 2 3 4 ]

CALCulate#:FEED 'XTIM:VOLT 1 | 2 | 3 | 4';MATH:STATe OFF;\*WAI

**Trace Coords**

[ LINEAR MAGNITUDE ]

CALCulate#:FORMat MLINear

[ LOG MAGNITUDE ]

CALC#:FORM MLIN::DISPlay:WINDow#:TRACe:Y:SPACing LOGarithmic

[ dB MAGNITUDE ]

CALCulate#:FORMat MLOGarithmic

[ PHASE ]

CALCulate#:FORMat PHASe

[ UNWRAPPED PHASE ]

CALCulate#:FORMat UPHase

[ MORE CHOICES ]

[ REAL PART ]

CALCulate#:FORMat REAL

[ IMAGINARY PART ]

CALCulate#:FORMat IMAGinary

[ NYQUIST DIAGRAM ]

CALCulate#:FORMat NYQuist

[ POLAR DIAGRAM ]

CALCulate#:FORMat POLar

[ GROUP DELAY ]

CALCulate#:FORMat GDElay

[ POLAR ROTATION ]

DISPlay:WINDow#:POLar:ROTation -360 ~ 360

[ CLOCKWISE ON OFF ]

DISPlay:WINDow#:POLar:CLOCKwise OFF | 0 | ON | 1

[ DELAY APERTURE ]

CALCulate#:GDAPerture:APERture 0 ~ 20[PCT]

[ X UNITS ]

[ HZ SEC ]

CALCulate#:UNIT:X 'HZ'

[ RPM SEC ]

CALCulate#:UNIT:X 'RPM'

[ ORDER REV ]

CALCulate#:UNIT:X 'ORD'

[ USER X UNIT ]

CALCulate#:UNIT:X 'USER'

Front Panel Key	GPIB Command
[ ORDER SETUP ]	
[ HZ/ORDER RATIO ]	CALCulate#:UNIT:X:ORDer:FACTor 1e-15 ~ 1e+15 [HZ/ORD   RPM/ORD]
[ TRACE RPM ]	CALCulate#:UNIT:X:ORDer:FACTor 1e-15 ~ 1e+15 [HZ/ORD   RPM/ORD]
[ USER X SETUP ]	
[ USER FREQ FACTOR ]	CALCulate#:UNIT:X:USER:FREQuency:FACTor 1e-15 ~ 1e+15
[ USER TIME FACTOR ]	CALCulate#:UNIT:X:USER:TIME:FACTor 1e-15 ~ 1e+15
[ FREQ LABEL ]	CALCulate#:UNIT:X:USER:FREQuency:LABel ' < STRING > '
[ TIME LABEL ]	CALCulate#:UNIT:X:USER:TIME:LABel ' < STRING > '
[ Y UNITS ]	
[ AMPLITUDE PK PP RMS ]	CALCulate#:UNIT:AMPLitude PEAK   PP   RMS
[ PHASE DEG RAD ]	CALCulate#:UNIT:ANGLe DEGRee   RADian
[ dB REF SETUP ]	
[ dBV dBEU ]	CALCulate#:UNIT:DBReference 'DBV'
[ dBm ]	CALCulate#:UNIT:DBReference 'DBM'
[ dBSPL 20uPa ]	CALCulate#:UNIT:DBReference 'DBSPL'
[ USER REFERENCE ]	CALCulate#:UNIT:DBReference 'USER'
[ dBm REF IMPEDANCE ]	CALCulate#:UNIT:DBReference:IMPedance 1e-15 ~ 1e+15[OHM]
[ USER REF LEVEL ]	CALCulate#:UNIT:DBReference REFeRence 1e-15 ~ 1e+15
[ USER LABEL ]	CALCulate#:UNIT:DBReference REFeRence:LABel ' < STRING > '
[ XDCR UNIT CONVERT ]	
[ g (ACCEL) ]	CALCulate#:UNIT:MECHanical 'G'
[ m/s^2 (ACCEL) ]	CALCulate#:UNIT:MECHanical 'M/S2'
[ m/s (VEL) ]	CALCulate#:UNIT:MECHanical 'M/S'
[ m (DISP) ]	CALCulate#:UNIT:MECHanical 'M'
[ inch/s^2 (ACCEL) ]	CALCulate#:UNIT:MECHanical 'INCH/S2'
[ inch/s (VEL) ]	CALCulate#:UNIT:MECHanical 'INCH/S'
[ inch (DISP) ]	CALCulate#:UNIT:MECHanical 'INCH'
[ mil (DISP) ]	CALCulate#:UNIT:MECHanical 'MIL'
[ VOLTS ]	CALCulate#:UNIT:VOLT 'V'
[ VOLTS^2 ]	CALCulate#:UNIT:VOLT 'V2'
[ V/rtHz ]	CALCulate#:UNIT:VOLT 'V/RTHZ'
[ V^2/Hz PSD ]	CALCulate#:UNIT:VOLT 'V2/HZ'
[ V^2s/Hz ESD ]	CALCulate#:UNIT:VOLT 'V2S/HZ'
[ X-AXIS LIN LOG ]	DISPlay:WINDow#:TRACe:X:SPACing LINear   LOGarithmic

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

Scale

[ AUTOSCALE ON OFF ]	DISPlay:WINDow#:TRACe:Y[:SCALe]AUTO OFF   0   ON   1   ONCE
[ TOP REFERENCE ]	DISPlay:WINDow#:TRACe:Y[:SCALe]:REFerence:TOP (-9.9e+37 ~ 9.9e+37[DBVRMS   VRMS   VPK   DBVPK   V   DBV   EU   DBEU])
[ CENTER REFERENCE ]	DISPlay:WINDow#:TRACe:Y[:SCALe]:REFerence:CENTer (-9.9e+37 ~ 9.9e+37[DBVRMS   VRSM   VPK   DBVPK   V   DBV   EU   DBEU])
[ BOTTOM REFERENCE ]	DISPlay:WINDow#:TRACe:Y[:SCALe]:REFerence:BOTTOm (-9.9e+37 ~ 9.9e+37[DBVRMS   VRMS   VPK   DBVPK   V   DBV   EU   DBEU])
[ INP RANGE TRACKING ]	DISPlay:WINDow#:TRACe:Y[:SCALe]:REFerence RANGe
[ Y PER DIV DECADES ]	DISPlay:WINDow#:TRACe:Y[:SCALe]PDIV (1e-06 ~ 9.9e+37[DB   VRMS   VPK   V])
[ MATCH X SCALE ]	DISPlay:WINDow#:TRACe:X:MATCh
[ MATCH Y SCALE ]	DISPlay:WINDow#:TRACe:Y:MATCh
[ FULL SCALE ]	DISPlay:WINDow#:TRACe:X[:SCALe]:AUTO ONCE DISPlay:WINDow :TRACe:Y[:SCALe]:AUTO ONCE

Active Trace

[ A ]	CALCulate#:ACTive A
[ B ]	CALCulate#:ACTive B
[ C ]	CALCulate#:ACTive C
[ D ]	CALCulate#:ACTive D
[ A B ]	CALCulate#:ACTive AB
[ C D ]	CALCulate#:ACTive CD
[ A B C D ]	CALCulate#:ACTive ABCD

Front Panel Key

GPIB Command

**Analys**

[ DEFINE FUNCTION ]

[ DEFINE F1 ]

CALCulate#:MATH:EXPRession1

[ MEAS DATA ]

[ PWR SPEC CHANNEL 1 2 3 4 ]

PSPEC1 | PSPEC2 | PSPEC3 | PSPECT4

[ + ]

+

[ - ]

-

[ \* ]

\*

[ / ]

/

[ ]

[ LIN SPEC CHANNEL 1 2 3 4 ]

LSPEC1 | LSPEC2 | LSPEC3 | LSPEC4

[ FREQ RESP ]

FRES21 | 31 | 41 | 43

[ COHERENCE ]

COH21 | 31 | 41 | 43

[ CROS SPEC ]

CSPEC21 | 31 | 41 | 43

[ TIME CHANNEL 1 2 3 4 ]

TIME1 | TIME2 | TIME3 | TIME4

[ WNDO TIME CHANNEL 1 2 3 4 ]

WTIME1 | WTIME2 | WTIME3 | WTIME4

[ RESAMPLED TIME CH 1 2 3 4 ]

TIME1 | TIME2 | TIME3 | TIME4

[ COMP PWR CHANNEL 1 2 3 4 ]

CPOW1 | CPOW2 | CPOW3 | CPOW4

[ ORDER TRK CHANNEL 1 2 3 4 ]

TRACK1 | TRACK2 | TRACK3 | TRACK4

[ RPM PROFILE ]

RPM

[ AUTO CORR CHANNEL 1 2 3 4 ]

ACORR1 | ACORR2 | ACORR3 | ACORR4

[ CROS CORR ]

XCORR21 | XCORR31 | XCORR41 | XCORR43

[ HISTOGRAM CHANNEL 1 2 3 4 ]

HIST1 | HIST2 | HIST3 | HIST4

[ PDF CHANNEL 1 2 3 4 ]

PDF1 | PDF2 | PDF3 | PDF4

[ CDF CHANNEL 1 2 3 4 ]

CDF1 | CDF2 | CDF3 | CDF4

[ DATA REG D1-D8 ]

[ DATA REG D1 ]

D1

[ DATA REG D2 ]

D2

[ DATA REG D3 ]

D3

[ DATA REG D4 ]

D4

[ DATA REG D5 ]

D5

[ DATA REG D6 ]

D6

[ DATA REG D7 ]

D7

[ DATA REG D8 ]

D8

[ CONSTANT K1-K5 ]

[ CONSTANT K1 ]

K1

[ CONSTANT K2 ]

K2

[ CONSTANT K3 ]

K3

[ CONSTANT K4 ]

K4

[ CONSTANT K5 ]

K5

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key	GPIB Command
[ FUNCTION F1-F5 ]	
[ FUNCTION F1 ]	F1
[ FUNCTION F2 ]	F2
[ FUNCTION F3 ]	F3
[ FUNCTION F4 ]	F4
[ FUNCTION F5 ]	F5
[ OPERATION ]	
[ CONJ( ) ]	CONJ(
[ MAG( ) ]	MAG(
[ REAL( ) ]	REAL(
[ IMAG( ) ]	IMAG(
[ SQRT( ) ]	SQRT(
[ FFT( ) ]	FFT(
[ INVERSE FFT( ) ]	IFFT(
[ PSD( ) ]	PSD(
[ MORE ]	
[ LN( ) ]	LN(
[ EXP( ) ]	EXP(
[ *jOMEGA( ) ]	XJOM(
[ /jOMEGA( ) ]	DJOM(
[ AWEIGHT( ) ]	AWEIGHT(
[ BWEIGHT( ) ]	BWEIGHT(
[ CWEIGHT( ) ]	CWEIGHT(
[ DIFF( ) ]	DIFF(
[ INTEG( ) ]	INTEG(
[ ]	(
[ DEFINE F2 ]	CALCulate#:MATH:EXPRession2
[ DEFINE F3 ]	CALCulate#:MATH:EXPRession3
[ DEFINE F4 ]	CALCulate#:MATH:EXPRession4
[ DEFINE F5 ]	CALCulate#:MATH:EXPRession5
[ DEFINE CONSTANT ]	
[ DEFINE K1 ]	CALCulate:MATH:CONStant1 -9.9e+37 ~ 9.9e+37 [,-9.9e+37 ~ 9.9e+37 ]
[ DEFINE K2 ]	CALCulate:MATH:CONStant2 -9.9e+37 ~ 9.9e+37 [,-9.9e+37 ~ 9.9e+37 ]
[ DEFINE K3 ]	CALCulate:MATH:CONStant3 -9.9e+37 ~ 9.9e+37 [,-9.9e+37 ~ 9.9e+37 ]
[ DEFINE K4 ]	CALCulate:MATH:CONStant4 -9.9e+37 ~ 9.9e+37 [,-9.9e+37 ~ 9.9e+37 ]
[ DEFINE K5 ]	CALCulate:MATH:CONStant5 -9.9e+37 ~ 9.9e+37 [,-9.9e+37 ~ 9.9e+37 ]
[ LIMIT TEST ]	
[ LINES ON OFF ]	DISPlay[:WINDow ]:LIMit:STATe OFF   0   ON   1
[ TEST EVAL ON OFF ]	CALCulate#:LIMit:STATe OFF   0   ON   1
[ FAIL BEEP ON OFF ]	CALCulate:LIMit:BEEP[:STATe] OFF   0   ON   1
[ DEFINE UPPER LIM ]	
[ START SEGMENT ]	
[ FINISH SEGMENT ]	CALCulate#:LIMit:UPPer:SEGMENT -9.9e+37 ~ 9.9e+37
[ MOVE ALL VERTICAL ]	CALCulate#:LIMit:UPPer:MOVE:Y -9.9e+37 ~ 9.9e+37
[ DELETE SEGMENT ]	CALCulate#:LIMit:UPPer:SEGMENT:CLEar

Front Panel Key	GPIB Command
[ DELETE ALL ]	
[ CONFIRM/ DELETE ]	CALCulate#:LIMit:UPPer:CLEar
[ TRACE TO LIMIT ]	CALCulate#:LIMit:UPPer:TRACe[:IMMediate]
[ DEFINE LOWER LIM ]	
[ START SEGMENT ]	
[ FINISH SEGMENT ]	CALCulate#:LIMit:LOWer:SEGMENT -9.9e+37 ~ 9.9e+37
[ MOVE ALL VERTICAL ]	CALCulate#:LIMit:LOWer:MOVE:Y -9.9e+37 ~ 9.9e+37
[ ]	
[ DELETE SEGMENT ]	CALCulate#:LIMit:LOWer:SEGMENT:CLEar
[ DELETE ALL ]	
[ CONFIRM/ DELETE ]	CALCulate#:LIMit:LOWer:CLEar
[ TRACE TO LIMIT ]	CALCulate#:LIMit:LOWer:TRACe[:IMMediate]
[ DATA TABLE ]	
[ TABLE ON OFF ]	DISPlay:WINDow#:DTABLE[:STATe] OFF   0   ON   1
[ MARKER ON OFF ]	DISPlay:WINDow#:DTABLE:MARKer[:STATe] OFF   0   ON   1
[ SELECT X ENTRY ]	CALCulate#:MARKer:DTABLE:X:SElect[:POINT] 1 ~ 50
[ INSERT X VALUE ]	CALCulate#:MARKer:DTABLE:X:INSert -9.9e+37 ~ 9.9e+37 [HZ   S   ORD   RPM   COUNT   AVG   V   VPK]
[ ENTRY LABEL ]	CALCulate#:MARKer:DTABLE:X:LABel ' < STING > '
[ DELETE X ]	CALCulate#:MARKer:DTABLE:X:DELeTe
[ CLEAR TABLE ]	CALCulate#:MARKer:DTABLE:CLEar[:IMMediate]
[ COPY TABLE ]	CALCulate#:MARKer:DTABLE:COpy[1   2   3   4]
[ Y VALUES ABS REL ]	CALCulate#:MARKer:MODE ABSolute   RELative
[ ABORT FIT ]	CALCulate2:CFIT:ABORt
[ CURVE FIT ]	
[ START FIT ]	CALCulate2:CFIT[:IMMediate]
[ ABORT FIT ]	CALCulate2:CFIT:ABORt
[ CURVE FIT REGISTER ]	
[ D1 ]	CALCulate#:CFIT:DESTination D1
[ D2 ]	CALCulate:CFIT:DESTination D2
[ D3 ]	CALCulate:CFIT:DESTination D3
[ D4 ]	CALCulate:CFIT:DESTination D4
[ D5 ]	CALCulate:CFIT:DESTination D5
[ D6 ]	CALCulate:CFIT:DESTination D6
[ D7 ]	CALCulate:CFIT:DESTination D7
[ D8 ]	CALCulate:CFIT:DESTination D8
[ COPY FROM SYNTHESIS ]	CALCulate:CFIT:COpy SYNThesis
[ FIT REGION ]	
[ FULL SPAN ]	CALCulate#:CFIT:FREQuency[:AUTO] ON
[ USER SPAN ]	CALCulate#:CFIT:FREQuency[:AUTO] OFF
[ START ]	CALCulate#:CFIT:FREQuency:STARt 0 ~ 115000[HZ]
[ STOP ]	CALCulate#:CFIT:FREQuency:STOP 0.390625 ~ 115000[HZ]

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

[ CURVE FIT SETUP ]	
[ ORDER MAX FIXED ]	CALCulate:CFIT:ORDER:AUTO OFF   0   ON   1
[ NUMBER OF POLES ]	CALCulate:CFIT:ORDER:POLES 0 ~ 20
[ NUMBER OF ZEROS ]	CALCulate:CFIT:ORDER:ZEROS 0 ~ 20
[ WEIGHT AUTO USER ]	CALCulate:CFIT:WEIGHT:AUTO OFF   0   ON   1
[ WEIGHT REGISTER ]	
[ D1 ]	CALCulate :CFIT:WEIGHT:REGister D1
[ D2 ]	CALCulate:CFIT:WEIGHT:REGister D2
[ D3 ]	CALCulate:CFIT:WEIGHT:REGister D3
[ D4 ]	CALCulate:CFIT:WEIGHT:REGister D4
[ D5 ]	CALCulate:CFIT:WEIGHT:REGister D5
[ D6 ]	CALCulate:CFIT:WEIGHT:REGister D6
[ D7 ]	CALCulate:CFIT:WEIGHT:REGister D7
[ D8 ]	CALCulate:CFIT:WEIGHT:REGister D8
[ TIME DELAY ]	CALCulate:CFIT:TDElay -100 ~ 100[S]
[ FREQUENCY SCALE ]	CALCulate:CFIT:FSCale 1e-06 ~ 1000000
[ TABLE ON OFF ]	DISPlay:VIEW CTAB
[ SYNTHESIS ]	
[ START SYNTHESIS ]	CALCulate:SYNThesis:[IMMediate]
[ SYNTHESIS REGISTER ]	
[ D1 ]	CALCulate#:SYNThesis:DESTination D1
[ D2 ]	CALCulate:SYNThesis:DESTination D2
[ D3 ]	CALCulate:SYNThesis:DESTination D3
[ D4 ]	CALCulate:SYNThesis:DESTination D4
[ D5 ]	CALCulate:SYNThesis:DESTination D5
[ D6 ]	CALCulate:SYNThesis:DESTination D6
[ D7 ]	CALCulate:SYNThesis:DESTination D7
[ D8 ]	CALCulate:SYNThesis:DESTination D8
[ COPY FROM CURVE FIT ]	CALCulate:SYNThesis:COPY CFIT
[ CONVERT TABLE ]	
[ CONVRT TO POLE ZERO ]	CALCulate :SYNThesis:TTYp# PZERo
[ CONVRT TO POLE RESD ]	CALCulate:SYNThesis:TTYPe PFRaction
[ CONVRT TO POLYNMIAL ]	CALCulate:SYNThesis:TTYPe POLYnomial
[ SYNTHESIS SETUP ]	
[ GAIN FACTOR ]	CALCulate2:SYNThesis:GAIN -9.9e37 ~ 9.9e37
[ TIME DELAY ]	CALCulate2:SYNThesis:TDElay -100 ~ 100[S]
[ FREQUENCY SCALE ]	CALCulate2:SYNThesis:FSCale 1e-06 ~ 1000000
[ X-AXIS LIN LOG ]	CALCulate2:SYNThesis:SPACing LINear   LOGarithmic
[ TABLE ON OFF ]	DISPlay:VIEW STAB



Front Panel Key

GPIB Command

**DispFormat**

[ SINGLE ]	DISPlay:FORMat SINGLE
[ SINGLE FRNT/ BACK ]	DISPlay:FORMat FBACk
[ UPPER/ LOWER ]	DISPlay:FORMat ULOWer
[ UPPER/BIG LOWER ]	DISPlay:FORMat SUBL
[ UPPR/LOWR FRNT/ BACK ]	DISPlay:FORMat ULFB
[ QUAD ]	DISPlay:FORMat QUAD
[ MEASURMNT STATE ]	DISPlay:VIEW MSTate
[ WATERFALL SETUP ]	
[ WATERFALL ON OFF ]	DISPlay[:WINDOW#]:WATerfall[:STATe] OFF   0   ON   1
[ Z-AXIS TOP ]	DISPlay[:WINDOW#]:WATerfall:TOP 0 ~ 9.9e+37[S   RPM   COUNT   AVG]
[ Z-AXIS BOTTOM ]	DISPlay[:WINDOW#]:WATerfall:BOTTom 0 ~ 9.9e+37[S   RPM   COUNT   AVG]
[ Z AXIS RANGE ]	DISPlay[:WINDOW#]:WATerfall:COUNT 1 ~ 50000
[ SKEW ON OFF ]	DISPlay[:WINDOW#]:WATerfall:SKEW OFF   0   ON   1
[ SKEW ANGLE ]	DISPlay[:WINDOW#]:WATerfall:SKEW:ANGLe 0 ~ 45
[ BASELINE SUPPRESS ]	DISPlay[:WINDOW#]:WATerfall:BASeline 0 ~ 100[PCT]
[ WATERFALL STEPS ]	CALCulate#:WATerfall:COUNT 1 ~ 32767
[ MORE ]	
[ TRACE HEIGHT ]	DISPlay[:WINDOW#]:WATerfall:HEIGHt 1 ~ 100[PCT]
[ HIDN LINE REMOVED ]	DISPlay[:WINDOW#]:WATerfall:HIDDen ON   1
[ HIDN LINE DISPLAYED ]	DISPlay[:WINDOW#]:WATerfall:HIDDen OFF   0 [ MORE ]
[ GRID ON OFF ]	DISPlay[:WINDow ]:TRACe:GRATicule:GRID[:STATe] OFF   0   ON   1
[ BODE DIAGRAM ]	DISPlay:BODE
[ TACH DISP ON OFF ]	DISPlay:RPM[:STATE] OFF   0   ON   1
[ TRACE TITLE ]	DISPlay[:WINDow ]TRACe:LABel ' < STRING > '
[ DFLT TITL ON OFF ]	DISPlay[:WINDow ]TRACe:LABel:DEFault[:STATe] OFF   0   ON   1
[ OCTAVE BND SETUP ]	
[ OVERALL ON OFF ]	DISPlay[:WINDow ]:TRACe:BPOWer[:STATe] OFF   0   ON   1
[ WEIGHTED ON OFF ]	DISPlay[:WINDow ]:TRACe:APOWer[:STATe] OFF   0   ON   1
[ MORE ]	
[ ANNOTATN ON OFF ]	DISPlay:ANNotation[:ALL] OFF   0   ON   1
[ DISPLAY ON OFF ]	DISPlay:ENABle OFF   0   ON   1
[ DISPLAY BRIGHTNESS ]	DISPlay:BRIGHtness 0.5 ~ 1
[ EXT MON ON OFF ]	DISPlay:EXTernal[:STATe] OFF   0   ON   1

## Marker Group

### Front Panel Key

### GPIB Command

#### Marker

[ MARKER ON OFF ]	CALCulate#:MARKer[:STATe] OFF   0   ON   1
[ COUPLED ON OFF ]	CALCulate#:MARKer:COUPled[:STATe] OFF   0   ON   1
[ MARKER X ENTRY ]	CALCulate#:MARKer:X[:ABSolute] -9.9e+37 ~ 9.9e+37 [HZ   S   VRMS   VPK   V   ORD   RPM]
[ MKR VALUE ABS REL ]	CALCulate#:MARKer:MODE ABSolute   RELative
[ REFERENCE TO MARKER ]	CALCulate#:MARKer:MODE RELative;REFerence:X (CALC#:MARK:X? ; Y CALC#:MARK:Y?)
[ REFERENCE SETUP ]	
[ REFERENCE TO MARKER ]	CALCulate#:MARKer:REFerence:MODE RELative;X CALCulate#:MARKer:X[:ABSolute] ; Y CALCulate#:MARKer:Y[:ABSolute]
[ REFERENCE X ENTRY ]	CALCulate#:MARKer:REFerence:X -9.9e+37 ~ 9.9e+37 [HZ   S   VRMS   VPK   V   ORD   RPM]
[ REFERENCE Y ENTRY ]	CALCulate#:MARKer:REFerence:Y -9.9e+37 ~ 9.9e+37 [DBVRMS   VPK   DBVPK   V   DBV   EU   DBEU   VRMS]
[ PEAK TRK ON OFF ]	CALCulate#:MARKer:MAXimum[:GLOBal]TRACk OFF   0   ON   1
[ NEXT PEAK RIGHT ]	CALCulate#:MARKer:MAXimum:RIGHT
[ NEXT PEAK LEFT ]	CALCulate#:MARKer:MAXimum:LEFT
[ MARKER TO PEAK ]	CALCulate#:MARKer:MAXimum[:GLOBal]

#### Marker Fctn

[ MARKER FCTN OFF ]	CALCulate#:MARKer:FUNCTion OFF
[ HARMONIC MARKER ]	
[ FUNDAMNTL FREQUENCY ]	CALCulate#:MARKer:HARMonic:FUNDamental (-9.9e+37 ~ 9.9e+37[HZ   ORD   RPM])
[ NUMBER OF HARMONICS ]	CALCulate#:MARKer:HARMonic:COUNT 0 ~ 400
[ COMPUTE OFF ]	CALCulate#:MARKer:FUNCTion OFF
[ THD ]	CALCulate#:MARKer:FUNCTion THD
[ HARMONIC POWER ]	CALCulate#:MARKer:FUNCTion HPOWER
[ MARKRS TO DATA TABL ]	CALCulate#:MARKer:DTABLE:X:SElect[:POINT] 1 ~ 50;INSert -9.9e+37 ~ 9.9e+37
[ HZ   S   ORD   RPM   COUNT   AVG   V   VPK ];LABel ' < STRING > '	
[ DATA TABL ON OFF ]	DISPlay:WINDow#:DTABLE[:STATe] OFF   0   ON   1
[ BAND MARKER ]	
[ BAND SPAN ]	CALCulate#:MARKer:BAND:SPAN 0 ~ 102400[HZ]
[ BAND CENTER ]	CALCulate#:MARKer:BAND:CENTer 0 ~ 102400[HZ]
[ BAND START ]	CALCulate#:MARKer:BAND:STARt (-9.9e+37 ~ 9.9e+3737[HZ   ORD   REV   RPM   S])
[ BAND STOP ]	CALCulate#:MARKer:BAND:STOP (-9.9e+37 ~ 9.9e+3737[HZ   ORD   REV   RPM   S])

Front Panel Key	GPIB Command
[ COMPUTE OFF ]	CALCulate#:MARKer:FUNCTion OFF
[ BAND POWER ]	CALCulate#:MARKer:FUNCTion BPOWer
[ RMS SQRT(PWR) ]	CALCulate#:MARKer:FUNCTion BRMS
[ SIDEBAND MARKER ]	
[ CARRIER FREQ ]	CALCulate#:MARKer:SIDeband:CARRier -9.9e+37 ~ 9.9e+37[HZ   ORD   RPM]
[ SIDEBAND INCREMENT ]	CALCulate#:MARKer:SIDeband:INCRe ment (-9.9e+37 ~ 9.9e+37[HZ   ORD   RPM])
[ NUMBER OF SIDEBANDS ]	CALCulate#:MARKer:SIDeband:COUNT 0 ~ 200
[ COMPUTE OFF ]	CALCulate#:MARKer:FUNCTion OFF
[ SIDEBAND POWER ]	CALCulate#:MARKer:FUNCTion SPOWer
[ MARKRS TO DATA TABL ]	CALCulate#:MARKer:DTABLE:X:SElect[:POINt] 1 ~ 50;INSert -9.9e+37 ~ 9.9e+37 [HZ   S   ORD   RPM   COUNT   AVG   V   VPK];LAbel ' < STRING > '
[ DATA TABL ON OFF ]	DISPlay:WINDow#:DTABLE[:STATe] OFF   0   ON   1
[ WATERFALL MARKERS ]	
[ TRACE SELECT ]	CALC#:WATerfall:TRACe:SElect -9.9e+37 ~ 9.9e+37[S   RPM   COUNT]
[ SLICE SELECT ]	CALC#:WATerfall:SLICe:SElect -9.9e+37 ~ 9.9e+37 [HZ   S   VRMS   VPK   V   ORD   RPM]
[ SAVE AND DISP DATA ]	CALC#:WATerfall:TRACe:SElect#; :CALC#:WATerfall:TRACe:COPI D1; :CALCulate1:FEED 'D1'
[ SAVE TO DATA REG ]	
[ SELECT SAVE REG ]	
[ D1 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D1
[ D2 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D2
[ D3 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D3
[ D4 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D4
[ D5 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D5
[ D6 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D6
[ D7 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D7
[ D8 ]	CALC#:WATerfall:TRACe:SElect#;;CALC#:WATerfall:TRACe:COPI D8
[ WATERFALL SETUP ]	
[ WATERFALL ON OFF ]	DISPlay[:WINDOW#]:WATerfall[:STATe] OFF   0   ON   1
[ Z-AXIS TOP ]	DISPlay[:WINDOW#]:WATerfall:TOP 0 ~ 9.9e+37[S   RPM   COUNT   AVG]
[ Z-AXIS BOTTOM ]	DISPlay[:WINDOW#]:WATerfall:BOTTom 0 ~ 9.9e+37[S   RPM   COUNT   AVG]
[ Z AXIS RANGE ]	DISPlay[:WINDOW#]:WATerfall:COUNT 1 ~ 50000
[ SKEW ON OFF ]	DISPlay[:WINDOW#]:WATerfall:SKEW OFF   0   ON   1
[ SKEW ANGLE ]	DISPlay[:WINDOW#]:WATerfall:SKEW:ANGLe 0 ~ 45
[ BASELINE SUPPRESS ]	DISPlay[:WINDOW#]:WATerfall:BASeline 0 ~ 100[PCT]
[ WATERFALL STEPS ]	CALCulate#:WATerfall:COUNT 1 ~ 32767
[ MORE ]	
[ TRACE HEIGHT ]	DISPlay[:WINDOW#]:WATerfall:HEIGHt 1 ~ 100[PCT]
[ HIDN LINE REMOVED ]	DISPlay[:WINDOW#]:WATerfall:HIDDen ON   1
[ HIDN LINE DISPLAYED ]	DISPlay[:WINDOW#]:WATerfall:HIDDen OFF   0

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

[ TIME PARAMETERS ]

[ START TIME ]

CALCulate#:MARKer:BAND:START -9.9e+37 ~ 9.9e+37[S | REC | PNT]

[ STOP TIME ]

CALCulate#:MARKer:BAND:STOP -9.9e+37 ~ 9.9e+37[S | REC | PNT]

[ COMPUTE OFF ]

CALCulate#:MARKer:FUNCTION OFF

[ OVERSHOOT ]

CALCulate#:MARKer:FUNCTION OVERshoot

[ RISE TIME ]

CALCulate#:MARKer:FUNCTION RTIME

[ SETTling TIME ]

CALCulate#:MARKer:FUNCTION STIME

[ DELAY TIME ]

CALCulate#:MARKer:FUNCTION DTIME

[ GAIN/PHAS MARGINS ]

[ START FREQUENCY ]

CALCulate#:MARKer:BAND:START 0 ~ 102400[HZ]

[ STOP FREQUENCY ]

CALCulate#:MARKer:BAND:STOP 0 ~ 102400[HZ]

[ COMPUTE OFF ]

CALCulate#:MARKer:FUNCTION OFF

[ COMPUTE MARGINS ]

CALCulate#:MARKer:FUNCTION GMARgin

[ FREQ & DAMPING ]

[ START FREQUENCY ]

CALCulate#:MARKer:BAND:START 0 ~ 102400[HZ]

[ STOP FREQUENCY ]

CALCulate#:MARKer:BAND:STOP 0 ~ 102400[HZ]

[ COMPUTE OFF ]

CALCulate#:MARKer:FUNCTION OFF

[ COMPUTE COEFFICIENT ]

CALCulate#:MARKer:FUNCTION FREQUENCY

[ SUPPLEMENTAL INFO ]

CALCulate#:MARKer:FUNCTION SINF

## System Group

### Front Panel Key

### GPIB Command

**Preset**

[ DO PRESET ]  
[ RECALL AUTOSTATE ]

SYSTem:PRESet  
MMEMory:LOAD:STATe #, ' < FILENAME > '

**BASIC**

[ RUN PROGRAM 1 ]  
[ RUN PROGRAM 2 ]  
[ RUN PROGRAM 3 ]  
[ RUN PROGRAM 4 ]  
[ RUN PROGRAM 5 ]  
[ DISPLAY SETUP ]

PROGram:[SElected]STATe PAUSE  
PROGram:NAME PROGAm1;STATe RUN  
PROGram:NAME PROGAm2;STATe RUN  
PROGram:NAME PROGAm3;STATe RUN  
PROGram:NAME PROGAm4;STATe RUN  
PROGram:NAME PROGAm5;STATe RUN

[ OFF ]  
[ FULL ]  
[ UPPER ]  
[ LOWER ]  
[ CLEAR SCREEN ]

DISPlay:PROGAm[:MODE] OFF  
DISPlay:PROGAm[:MODE] FULL  
DISPlay:PROGAm[:MODE] UPPer  
DISPlay:PROGAm[:MODE] LOWer  
DISPlay:ENABle OFF  
PROGAm[:SElected]:STATe CONTinue

[ CONTINUE ]  
[ INSTRUMNT BASIC ]

PROGAm[:SElected]:STATe RUN

[ RUN PROGRAM ]  
[ SELECT PROGRAM ]  
[ PROGRAM 1 ]  
[ PROGRAM 2 ]  
[ PROGRAM 3 ]  
[ PROGRAM 4 ]  
[ PROGRAM 5 ]

PROGAm[:SElected]:NAME PROG1  
PROGAm[:SElected]:NAME PROG2  
PROGAm[:SElected]:NAME PROG3  
PROGAm[:SElected]:NAME PROG4  
PROGAm[:SElected]:NAME PROG5  
PROGAm[:SElected]:LABel ' < STRING > '

[ LABEL PROGRAM ]  
[ PRINT PROGRAM ]

[ UTILITIES ]  
[ MEMORY SIZE ]  
[ AUTO MEMORY ]  
[ SCRATCH A ]  
[ PERFORM SCRATCH ]

PROGAm[:SElected]:MALLocate 1200 ~ 7340030  
PROGAm[:SElected]:MALLocate DEFault

[ DEBUG ]  
[ RUN ]  
[ CONTINUE ]  
[ SINGLE STEP ]  
[ LAST ERROR ]  
[ EXAMINE VARIABLE ]

PROGAm[:SElected]:DELeTe[:SElected]  
PROGAm[:SElected]:STATe RUN  
PROGAm[:SElected]:STATe CONTinue

[ RESET ]

PROGAm[:SElected]:NUMBER  
PROGAm[:SElected]:STRING  
PROGAm[:SElected]:STATe STOP

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key

GPIB Command

Save/Recall

[ SAVE DATA ]

[ SAVE TRACE ]

[ INTO D1 ]

TRACe:DATA D1,TRACe#

[ INTO D2 ]

TRACe:DATA D2,TRACe#

[ INTO D3 ]

TRACe:DATA D3,TRACe#

[ INTO D4 ]

TRACe:DATA D4,TRACe#

[ INTO D5 ]

TRACe:DATA D5,TRACe#

[ INTO D6 ]

TRACe:DATA D6,TRACe#

[ INTO D7 ]

TRACe:DATA D7,TRACe#

[ INTO D8 ]

TRACe:DATA D8,TRACe#

[ INTO FILE ]

MMEMoRY:STORe:TRACe#, '<FILENAME>'

[SAVE CAPTURE ]

MMEMoRY:STORe:TCAPture '<FILENAME>'

[SAVE WATERFALL ]

[ INTO W1 ]

TRACe:WATerfall:DATA W1 TRACe#

[ INTO W2 ]

TRACe:WATerfall:DATA W2 TRACe#

[ INTO W3 ]

TRACe:WATerfall:DATA W3 TRACe#

[ INTO W4 ]

TRACe:WATerfall:DATA W4 TRACe#

[ INTO W5 ]

TRACe:WATerfall:DATA W5 TRACe#

[ INTO W6 ]

TRACe:WATerfall:DATA W6 TRACe#

[ INTO W7 ]

TRACe:WATerfall:DATA W7 TRACe#

[ INTO W8 ]

TRACe:WATerfall:DATA W8 TRACe#

[ INTO FILE ]

MMEMoRY:STORe:WATerfall TRACe#,, '<FILENAME>'

[CONTINUE SAVE ]

MMEMoRY:STORe:CONTInue

[ CATALOG ON OFF ]

DISPlay:VIEW MMEMoRY

[ SAVE STATE ]

MMEMoRY:STORe:STATe #, '<FILENAME>'

[ SAVE MORE ]

[ SAVE UPPER LIM ]

MMEMoRY:STORe:LIMit:UPPer:TRACe# '<FILENAME>'

[ SAVE LOWER LIM ]

MMEMoRY:STORe:LIMit:LOWer:TRACe# '<FILENAME>'

[ SAVE MATH ]

MMEMoRY:STORe:MATH '<FILENAME>'

[ SAVE DATA TABLE ]

MMEMoRY:STORe:DTAbLe:TRACe[1 | 2 | 3 | 4] '<FILENAME>'

[ SAVE PROGRAM ]

MMEMoRY:STORe:PROGAm '<FILENAME>'

[ SAVE PROGRAM ASCII BIN ]

MMEMoRY:STORe:PROGAm:FORMat ASCII | BIN

[ SAVE FIT TABLE ]

MMEMoRY:STORe:CFIT '<FILENAME>'

[ SAVE SNTH TABLE ]

MMEMoRY:STORe:SYNTHeSis '<FILENAME>'

[ SAVE AUTOSTATE ]

MMEMoRY:STORe:STATe 1, 'NVRAM:AUTO\_ST'

Front Panel Key	GPIB Command
[ RECALL DATA ]	
[RECALL TRACE ]	
[ FROM FILE INTO D1 ]	MMEMory:LOAD:TRACe D1, '< FILENAME >'
[ FROM FILE INTO D2 ]	MMEMory:LOAD:TRACe D2, '< FILENAME >'
[ FROM FILE INTO D3 ]	MMEMory:LOAD:TRACe D3, '< FILENAME >'
[ FROM FILE INTO D4 ]	MMEMory:LOAD:TRACe D4, '< FILENAME >'
[ FROM FILE INTO D5 ]	MMEMory:LOAD:TRACe D5, '< FILENAME >'
[ FROM FILE INTO D6 ]	MMEMory:LOAD:TRACe D6, '< FILENAME >'
[ FROM FILE INTO D7 ]	MMEMory:LOAD:TRACe D7, '< FILENAME >'
[ FROM FILE INTO D8 ]	MMEMory:LOAD:TRACe D8, '< FILENAME >'
[RECALL CAPTURE ]	MMEMory:LOAD:TCAPture '< FILENAME >'
[ RECALL WATERFALL ]	
[ FROM FILE INTO W1 ]	MMEMory:LOAD:WATerfall W1, '< FILENAME >'
[ FROM FILE INTO W2 ]	MMEMory:LOAD:WATerfall W2, '< FILENAME >'
[ FROM FILE INTO W3 ]	MMEMory:LOAD:WATerfall W3, '< FILENAME >'
[ FROM FILE INTO W4 ]	MMEMory:LOAD:WATerfall W4, '< FILENAME >'
[ FROM FILE INTO W5 ]	MMEMory:LOAD:WATerfall W5, '< FILENAME >'
[ FROM FILE INTO W6 ]	MMEMory:LOAD:WATerfall W6, '< FILENAME >'
[ FROM FILE INTO W7 ]	MMEMory:LOAD:WATerfall W7, '< FILENAME >'
[ FROM FILE INTO W8 ]	MMEMory:LOAD:WATerfall W8, '< FILENAME >'
[ CONTINUE RECALL ]	MMEMory:LOAD:CONTinue
[ RECALL STATE ]	MMEMory:LOAD:STATe #, '< FILENAME >'
[ RECALL MORE ]	
[ RECALL UPPER LIM ]	MMEMory:LOAD:LIMit:UPPer:TRACe# '< FILENAME >'
[ RECALL LOWER LIM ]	MMEMory:LOAD:LIMit:LOWer:TRACe# '< FILENAME >'
[ RECALL MATH ]	MMEMory:LOAD:MATH '< FILENAME >'
[ RECALL DATA TABL ]	MMEMory:LOAD:DTABle:TRACe[1   2   3   4] '< FILENAME >'
[ RECALL PROGRAM ]	MMEMory:LOAD:PROGram '< FILENAME >'
[ RCL FIT TABLE ]	MMEMory:LOAD:CFIT '< FILENAME >'
[ RCL SYNTH TABLE ]	MMEMory:LOAD:SYNTHeSis '< FILENAME >'
[ RECALL AUTOSTATE ]	MMEMory:LOAD:STATe 1, 'NVRAM:AUTO_ST'
[ DEFAULT DISK ]	
[ NON-VOL RAM DISK ]	MMEMory:MSIS 'NVRAM:'
[ VOLATILE RAM DISK ]	MMEMory:MSIS 'RAM:'
[ INTERNAL DISK ]	MMEMory:MSIS 'INT:'
[ EXTERNAL DISK ]	MMEMory:MSIS 'EXT:'
[ IEXTERNAL DISK UNIT ]	MMEMory:DISK:UNIT 0 ~ 10
[ CREATE DIRECTORY ]	MMEMory:MDIRECTORY '< STRING >'
[ DELETE DIRECTORY ]	MMEMory:DELete < MMEMNAME >





Front Panel Key

GPIB Command

**Plot/Print**

[ START PLOT/PRNT ]	HCOPY[:IMMEDIATE]
[ PLOT/PRNT DEVICE ]	
[HP-GL PLOTTER ]	HCOPY:DEVICE:LANGUAGE HPGL
[RASTER PRINTER ]	HCOPY:DEVICE:LANGUAGE PCL
[HP-GL PRINTER ]	HCOPY:DEVICE:LANGUAGE PHPG
[ PLOT/PRNT DESTINATN ]	
[OUTPUT TO GPIB ]	HCOPY:DESTINATION 'SYSTEM:COMMUNICATE:GPIB:RDEV'
[OUTPUT TO SERIAL ]	HCOPY:DESTINATION 'SYSTEM:COMMUNICATE:SERIAL'
[OUTPUT TO PARALLEL ]	HCOPY:DESTINATION 'SYSTEM:COMMUNICATE:CENTRONICS'
[OUTPUT TO FILE ]	HCOPY:DESTINATION 'MMEM'
[ OUTPUT FILENAME ]	MMEMORY:NAME ' < FILENAME > '
[ PLOT DATA SELECT ]	
[ALL ]	HCOPY:ITEM:ALL[:IMMEDIATE]
[TRACE ]	HCOPY:ITEM[:WINDOW#]:TRACE[:IMMEDIATE]
[TRACE MARKER ]	HCOPY:ITEM[:WINDOW#]:TRACE:MARKER[:IMMEDIATE]
[MARKER REFERENCE ]	HCOPY:ITEM[:WINDOW#]:TRACE:MARKER:REFERENCE[:IMMEDIATE]
[GRID ]	HCOPY:ITEM[:WINDOW#]:TRACE:GRATICULE[:IMMEDIATE]
[ PLOT PEN SETUP ]	
[DEFAULT PENS ]	HCOPY:COLOR:DEFAULT
[TRACE A PEN ]	HCOPY:ITEM:WINDOW1:TRACE:COLOR 0 ~ 16
[TRACE B PEN ]	HCOPY:ITEM:WINDOW2:TRACE:COLOR 0 ~ 16
[TRACE C PEN ]	HCOPY:ITEM:WINDOW3:TRACE:COLOR 0 ~ 16
[TRACE D PEN ]	HCOPY:ITEM:WINDOW4:TRACE:COLOR 0 ~ 16
[ MARKER PEN SETUP ]	
[TRACE A MKR PEN ]	HCOPY:ITEM:WINDOW1:TRACE:MARKER:COLOR 0 ~ 16
[TRACE B MKR PEN ]	HCOPY:ITEM:WINDOW2:TRACE:MARKER:COLOR 0 ~ 16
[TRACE C MKR PEN ]	HCOPY:ITEM:WINDOW3:TRACE:MARKER:COLOR 0 ~ 16
[TRACE D MKR PEN ]	HCOPY:ITEM:WINDOW4:TRACE:MARKER:COLOR 0 ~ 16
[ALPHA PEN ]	HCOPY:ITEM:LABEL:COLOR 0 ~ 16
[GRID PEN ]	HCOPY:ITEM:WINDOW1:TRACE:GRATICULE:COLOR 0 ~ 16
[ PLOT LINE SETUP ]	
[TRACE A LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDOW1:TRACE:LTYPE SOLID
[ DOTTED ]	HCOPY:ITEM:WINDOW1:TRACE:LTYPE DOTTED
[ DASHED ]	HCOPY:ITEM:WINDOW1:TRACE:LTYPE DASHED
[ USER LINE TYPE ]	HCOPY:ITEM:WINDOW1:TRACE:LTYPE -6 ~ 6
[TRACE B LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDOW2:TRACE:LTYPE SOLID
[ DOTTED ]	HCOPY:ITEM:WINDOW2:TRACE:LTYPE DOTTED
[ DASHED ]	HCOPY:ITEM:WINDOW2:TRACE:LTYPE DASHED
[ USER LINE TYPE ]	HCOPY:ITEM:WINDOW2:TRACE:LTYPE -6 ~ 6

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key	GPIB Command
[TRACE C LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow3:TRACe:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow3:TRACe:LTYPe DOTTed
[ DASHED ]	HCOPY:ITEM:WINDow3:TRACe:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow3:TRACe:LTYPe -6 ~ 6
[TRACE D LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow4:TRACe:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow4:TRACe:LTYPe DOTTed
[ DASHED ]	HCOPY:ITEM:WINDow4:TRACe:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow4:TRACe:LTYPe -6 ~ 6
[LIMIT A LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow1:TRACe:LIMit:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow1:TRACe:LIMit:LTYPe DOTted
[ DASHED ]	HCOPY:ITEM:WINDow1:TRACe:LIMit:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow1:TRACe:LIMit:LTYPe -6 ~ 6
[LIMIT B LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow2:TRACe:LIMit:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow2:TRACe:LIMit:LTYPe DOTted
[ DASHED ]	HCOPY:ITEM:WINDow2:TRACe:LIMit:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow2:TRACe:LIMit:LTYPe -6 ~ 6
[LIMIT C LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow3:TRACe:LIMit:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow3:TRACe:LIMit:LTYPe DOTted
[ DASHED ]	HCOPY:ITEM:WINDow3:TRACe:LIMit:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow3:TRACe:LIMit:LTYPe -6 ~ 6
[LIMIT D LINE TYPE ]	
[ SOLID ]	HCOPY:ITEM:WINDow4:TRACe:LIMit:LTYPe SOLid
[ DOTTED ]	HCOPY:ITEM:WINDow4:TRACe:LIMit:LTYPe DOTted
[ DASHED ]	HCOPY:ITEM:WINDow4:TRACe:LIMit:LTYPe DASHed
[ USER LINE TYPE ]	HCOPY:ITEM:WINDow4:TRACe:LIMit:LTYPe -6 ~ 6
[ MORE SETUP ]	
[ PLOT PEN SPEED ]	
[ FAST 50 cm/s ]	HCOPY:DEVice:SPEed 50
[ SLOW 10 cm/s ]	HCOPY:DEVice:SPEed 10
[ DEFINE ? cm/s ]	HCOPY:DEVice:SPEed 1 ~ 100
[ P1 P2 SETUP ]	
[ USER P1 P2 ON OFF ]	HCOPY:PAGE:DIM:AUTO OFF   0   ON   1
[ USER P1 X ]	
[ USER P1 Y ]	HCOPY:PAGE:DIM:USER:LLEF -32767 ~ 32767
[ USER P2 X ]	
[ USER P2 Y ]	HCOPY:PAGE:DIM:USER:URIG -32767 ~ 32767
[ TIME STMP ON OFF ]	HCOPY:ITEM:TSTamp[:STATe] OFF   0   ON   1
[PAGE EJCT ON OFF ]	HCOPY:EJECt OFF   0   ON   1
[ TITLE LINE 1 ]	HCOPY:TITL1 ' < STRING > '
[ TITLE LINE 2 ]	HCOPY:TITL2 ' < STRING > '

Front Panel Key

GPIB Command

[ SERIAL SETUP ]  
 [ BAUD RATE ]  
 [ BITS/CHAR ]  
 [ STOP BITS ]  
 [ PARITY ]  
 [ PRY CHK ]  
 [ RCVR PACE ]  
 [ XMIT PACE ]

SYSTem:COMMunicate:SERial[:RECeive]:BAUD 300 ~ 9600  
 SYSTem:COMMunicate:SERial[:RECeive]:BITS 5 ~ 8  
 SYSTem:COMMunicate:SERial[:RECeive]:SBITS 1 ~ 2  
 SYSTem:COMMunicate:SERial[:RECeive]:PARITY[:TYPE] NONE | EVEN | ODD  
 SYSTem:COMMunicate:SERial[:RECeive]:PARITY:CHECK OFF | 0 | ON | 1  
 SYSTem:COMMunicate:SERial[:RECeive]:PACE NONE | XON  
 SYSTem:COMMunicate:SERial:TRANsmit:PACE NONE | XON | DSR

**System**

Utility

[ SHOW LAST MESSAGES ]

DISPlay:VIEW MESS

[ CALIBRATN ]

[AUTO CAL ON OFF ]  
 [SINGLE CAL ]

CALibration:AUTO OFF | 0 | ON | 1 | ONCE  
 CALibration:AUTO ONCE

[SAVE CH1 CAL TRACE ]

[ INTO D1 ]  
 [ INTO D2 ]  
 [ INTO D3 ]  
 [ INTO D4 ]  
 [ INTO D5 ]  
 [ INTO D6 ]  
 [ INTO D7 ]  
 [ INTO D8 ]

TRACe:DATA D1,CAL1  
 TRACe:DATA D2,CAL1  
 TRACe:DATA D3,CAL1  
 TRACe:DATA D4,CAL1  
 TRACe:DATA D5,CAL1  
 TRACe:DATA D6,CAL1  
 TRACe:DATA D7,CAL1  
 TRACe:DATA D8,CAL1

[SAVE CH2 CAL TRACE ]

[ INTO D1 ]  
 [ INTO D2 ]  
 [ INTO D3 ]  
 [ INTO D4 ]  
 [ INTO D5 ]  
 [ INTO D6 ]  
 [ INTO D7 ]  
 [ INTO D8 ]

TRACe:DATA D1,CAL2  
 TRACe:DATA D2,CAL2  
 TRACe:DATA D3,CAL2  
 TRACe:DATA D4,CAL2  
 TRACe:DATA D5,CAL2  
 TRACe:DATA D6,CAL2  
 TRACe:DATA D7,CAL2  
 TRACe:DATA D8,CAL2

[SAVE CH3 CAL TRACE ]

[ INTO D1 ]  
 [ INTO D2 ]  
 [ INTO D3 ]  
 [ INTO D4 ]  
 [ INTO D5 ]  
 [ INTO D6 ]  
 [ INTO D7 ]  
 [ INTO D8 ]

TRACe:DATA D1,CAL3  
 TRACe:DATA D2,CAL3  
 TRACe:DATA D3,CAL3  
 TRACe:DATA D4,CAL3  
 TRACe:DATA D5,CAL3  
 TRACe:DATA D6,CAL3  
 TRACe:DATA D7,CAL3  
 TRACe:DATA D8,CAL3

Cross-Reference from Front-Panel Keys  
to GPIB Commands

Front Panel Key	GPIB Command
[SAVE CH4 CAL TRACE]	
[ INTO D1 ]	TRACe:DATA D1,CAL4
[ INTO D2 ]	TRACe:DATA D2,CAL4
[ INTO D3 ]	TRACe:DATA D3,CAL4
[ INTO D4 ]	TRACe:DATA D4,CAL4
[ INTO D5 ]	TRACe:DATA D5,CAL4
[ INTO D6 ]	TRACe:DATA D6,CAL4
[ INTO D7 ]	TRACe:DATA D7,CAL4
[ INTO D8 ]	TRACe:DATA D8,CAL4
[ BEEPER ON OFF ]	SYSTem:BEEPer:STATe OFF 0 ON 1
[ CLOCK SETUP ]	
[TIME HHMM ]	SYSTem:TIME 0 ~ 2359
[DATE MMDDYY ]	SYSTem:DATE 10100 ~ 123199
[TIMESTAMP SETUP ]	
[ 24 HR DD/MM/YY ]	HCOPy:TSTamp:MODE CHOice1
[ 24 HR DD.MM.YY ]	HCOPy:TSTamp:MODE CHOice2
[ 24 HR YY MM DD ]	HCOPy:TSTamp:MODE CHOice3
[ 12 HR DD/MM/YY ]	HCOPy:TSTamp:MODE CHOice4
[ 12 HR MM-DD-YY ]	HCOPy:TSTamp:MODE CHOice5
[ OPTIONS SETUP ]	DISPlay:VIEW OPT
[ MEMORY USAGE ]	DISPlay:VIEW MEM
[REMOVE CAPTURE ]	
[ CONFIRM REMOVE ]	MEMory:DELeTe[:NAME] TCAPture
[REMOVE WATERFALL ]	
[ CONFIRM REMOVE ]	MEMory:DELeTe[:NAME] WATerfall
[REMOVE WTFL REGS ]	
[ CONFIRM REMOVE ]	MEMory:DELeTe[:NAME] WREGister
[REMOVE PROGRAMS ]	
[ CONFIRM REMOVE ]	MEMory:DELeTe[:NAME] PROGram
[REMOVE RAM DISK ]	
[ CONFIRM REMOVE ]	MEMory:DELeTe[:NAME] RDISK
[ FAN SETUP ]	
[ FAN OFF ]	SYSTem:FAN OFF
[ FAN FULL SPEED ]	SYSTem:FAN FULL
[ AUTOMATIC SPEED ]	SYSTem:FAN AUTO
[ MORE ]	
[ FAULT LOG ]	DISPlay:VIEW FTABle
[CLEAR FAULT LOG ]	SYSTem:FLOG:CLEAr
[ SELF TEST ]	
[ QUICK CONF TEST ]	*TST?
[ LONG CONF TEST ]	TEST:LONG
[ TEST LOG ]	DISPlay:VIEW TTAB
[ CLEAR TEST LOG ]	TEST:LOG:CLEAr
[ S/N VERSION ]	*IDN?

## **Error Messages**

---



## Error Messages

---

### Introduction

This appendix contains a listing of all the error messages that can be generated by the HP 35670A in response to GPIB commands. Each message consists of an error number (always negative) followed by a string. The string contains a general description of the error followed by additional information about the cause of the error.

In this appendix, error numbers and their general descriptions are shown using a bold font. Phrases that complete the descriptions with additional information are grouped under the associated error number.

Up to five error messages are temporarily stored in the analyzer's error queue. They are returned to the controller, one message at a time, when you send the SYST:ERR query.

## Command Errors

**-100: Command error.**

Command is query only.  
Too large blocksize required.

**-104: Parameter not allowed.**

**-109: Missing parameter.**

Missing parameter.  
Parameter not allowed.

**-113: Undefined header.**

**-120: Illegal parameter.**

**-131: Invalid suffix.**

**-141: Invalid character data.**

**-151: Invalid string data.**

**-161: Invalid block data.**

Invalid block data.  
Data block does not contain a STATE.  
Data block does not contain valid MATH definitions.



---

## Execution Errors

### -200: Execution error.

Amplitude selections are meaningless for the current combination of Meas Data and Trace Coord selections.  
Blank lines or Laurent terms cannot be changed, deleted or undeleted.  
Bode diagram available in 2 or 4 channel FFT ANALYSIS, SWEPT SINE, or ORDER ANALYSIS with TRACK ON.

CAPTURE ABORTED !!

A measurement parameter was changed.

Pressed [ ABORT CAPTURE ] softkey.

Specified maximum RPM for tachometer < actual RPM; maximum RPM is too small.

Can't move the marker reference to the power band.

Cannot continue capture playback.

Cannot continue measurement Measurement state has changed or Calibration has run Press START to take new data (existing data will be lost).

Cannot edit an empty data register.

Capture cannot be used with Inst Mode SWEPT SINE.

Channel 1 input cannot be disabled.

Channel 2 trigger only available with two or more active channels (Inst Mode, CHANNELS 2 or CHANNELS 4).

Constant trace. No frequency and damping.

Conversion failure. Double precision overflow.

Curve fit may be poor. Coherence around peaks in weighting function is too low.

Data Tables cannot be turned ON with FRNT/BACK display formats.

Delete program Not Allowed while RECORDING ENABLED.

Does not match marker units.

Double integrations are not allowed on time domain data.

Download program not Allowed while RECORDING ENABLED.

Exponent overflow. E+308 limit exceeded.

Exponent underflow. E-308 limit exceeded.

[Function register] definition is not valid for execution.

[Function register] execution requires recursion.

[Function register] not defined.

FREQUENCY RESPONSE, COHERENCE, and CROSS SPECTRUM data are not valid with

PEAK HOLD average.

Fell out of real time data acquisition. .

File operation aborted, Capture changed.

File operation aborted, Capture in progress.

File operation aborted. Waterfall changed.

File operation aborted. Waterfall register changed.

File operation not completed.

Fit aborted. Curve fit algorithm failed. Must change input to curve fit.

Fit aborted. Trace A data cannot be Octave.

Fit aborted. Trace A data invalid.

Fit aborted. Trace A data must be 32-bit floating point.

Fit aborted. Trace A data must be LINEAR or LOG spaced.

## Error Messages

### Execution Errors

Fit aborted. Trace A data must be complex.

Fit aborted. Trace A data must be in frequency domain.

Fit aborted. Trace A is a constant. Does not have finite poles or zeros.

Fit aborted. User span completely outside trace A boundaries.

Fit aborted. Weight data cannot be negative.

Fit aborted. Weight register data invalid.

Fit or Synth table data invalid.

Fit or Synth table data invalid. Found an invalid floating point value.

Fit or Synth table data invalid. System order greater than 20.

Fit or Synth table data invalid. Table value outside valid range.

Fit table format invalid. Only pole zero format allowed.

Function definition contains references to Meas Data selections that are not available in this Inst Mode.

Function definition is not valid.

Function definition is too long.

Function definition may not reference higher numbered functions.

GPIB control not received.

Initial RPM [RPM value] < Min RPM.

Initial RPM [RPM value] > Max RPM.

Input Range tracking not valid on this data.

Instrument BASIC not installed.

Instrument must be in either FFT ANALYSIS or SWEPT SINE mode.

Invalid Function Code.

Invalid Instrument State Request.

Invalid Instrument State Value.

Invalid Instrument State.

Invalid TRACE COORD selection.

LOG X AXIS invalid with NYQUIST or ORBIT.

Limit table invalid.

Limit testing is turned off.

Limits are not allowed on ORBITS, NYQUIST or POLAR Trace Coordinates, or data with arbitrarily spaced x axis values.

Limits are undefined.

Limits not allowed with WATERFALL ON.

MEAS DATA selection not available with Inst Mode CORRELATN ANALYSIS.

MEAS DATA selection not available with Inst Mode FFT ANALYSIS.

MEAS DATA selection not available with Inst Mode HISTOGRAM.

MEAS DATA selection not available with Inst Mode OCTAVE ANALYSIS.

MEAS DATA selection not available with Inst Mode ORDER ANALYSIS.

MEAS DATA selection not available with Inst Mode SWEPT SINE.

MEAS DATA selection only available with four active channels (Inst Mode, CHANNELS 4).

MEAS DATA selection only available with two or more active channels (Inst Mode, CHANNELS 2 or CHANNELS 4).

MEAS DATA selection requires TRACK OFF (Key Path: [Freq]).

MEAS DATA selection requires TRACK ON (Key Path: [Freq]).

Marker Function invalid for non frequency domain MEASurement DATA.

Marker Function invalid for non time domain MEASurement DATA.

Marker Function invalid with WATERFALL ON.

Math not valid. [Specified math] operation cannot process a data block this large.

Math not valid. [Specified math] operation requires [specified measurement] data.  
NYQUIST and POLAR not available with WATERFALL ON.  
No CAPTURE data.  
No CAPTURE data for channel 2.  
No Coupled Markers allowed with WATERFALL ON.  
No Main Marker allowed with WATERFALL displays.  
No Marker Functions allowed with ORBIT Meas Data or NYQUIST or POLAR Trace Coord.  
No REFERENCE indicator allowed with WATERFALL displays.  
No REFERENCE indicator allowed with ORBIT Meas Data or NYQUIST or POLAR Trace Coord.  
Not a valid serial number.  
Not enough CAPTURE data for any Measurement Result.  
OPTION 1D0, Computed Order Tracking not installed.  
OPTION 1D1, Realtime Octave Measurements not installed.  
OPTION 1D2, Swept Sine Measurements not installed.  
OPTION 1D3 Curve Fit/Synthesis not installed.  
OPTION 1D4 Arbitrary Source not installed.  
ORBIT not available in zoom mode (Freq START not equal to 0 Hz).  
ORBIT not available with WATERFALL ON.  
Online measurement not possible. Decrease MaxOrder/DeltaOrder ratio or use Time Capture Playback.  
Only AUTOMATIC ARM can be used with EXTERNAL TRIGGER in ORDER ANALYSIS.  
Only AUTOMATIC ARM can be used with averaging in ORDER ANALYSIS.  
Pause the measurement before plotting a waterfall.  
Plot/Print already in progress.  
Plotter/printer not responding.  
Printer/Plotter is not on line.  
Printer/Plotter out of paper.  
Printer/Plotter reports error.  
Program memory re-size Not Allowed while RECORDING ENABLED.  
Program variable access Not Allowed while RECORDING ENABLED.  
RPM or ramp rate too high.  
Received GPIB control without requesting it.  
Recording mode canceled because: Instrument BASIC execution error. Refer to Instrument BASIC Users Handbook, Appendix A, "Error Messages."  
Reference position and value cannot be changed with Trace Coord set to POLAR.  
SAVE/RECALL PROGRAM Not Allowed during power-on calibration.  
SAVE/RECALL PROGRAM Not Allowed while RECORDING ENABLED.  
Select AVERAGE ON and AVERAGE TYPE RMS or RMS EXPONENTIAL to view COHERENCE data.  
Serial number must be 10 characters.  
Synth table data invalid. Complex polynomial coefficients not allowed.  
Synth table data invalid. Complex residue over real pole not allowed.  
Synth table data invalid. Need a residue for every pole.  
TIME CAPTURE data not available with WATERFALL ON.  
Table invalid. Complex coefficients not allowed.  
Table invalid. Complex residue over real pole not allowed.  
Table invalid. Need a residue for every pole. .  
Table order too large.  
The Instrument BASIC editor has been disabled.  
The imaginary term cannot be negative.

## Error Messages

### Execution Errors

The maximum order for this column has been reached. .

The total number of fixed poles or zeros in the table cannot be greater than the number entered under the curve fit setup key.

The value entered for number of poles cannot be less than the total fixed poles in the table.

The value entered for number of zeros cannot be less than the total fixed zeros in the table.

This data register contains data with non-uniform X-axis spacing and therefore cannot be edited.

This feature cannot be used when the marker is at zero.

This key is only valid when XDCR units are ON and the Y-axis of the active trace is an amplitude.

This marker function is not valid for arbitrarily spaced data.

This marker function is only valid for complex data.

This marker function is only valid for frequency domain data.

This marker function is only valid for frequency or order domain data.

This marker function is only valid for frequency, order, or octave domain data.

This marker function is only valid for frequency-response data.

This marker function is only valid for octave domain data.

This marker function is only valid for power or linear spectrum Trace Data and MAGNITUDE, REAL, or IMAGINARY Trace Coordinates.

This marker function is only valid for time domain data.

This menu is only valid for single channel frequency or time data when XDCR unit is ON and is not set to USER.

To use a 102.4 kHz span the antialias filter must be turned off for channel 1.

To use the TRACE RPM feature, data must be taken in RPM STEP ARM mode.

Too many rpm steps for order tracking.

Trace contains invalid data.

Trace has too many points to convert to a limit line, only the first 1024 points will be used.

UP ONLY auto range not available in Swept Sine instrument mode.

Unit Conversion not possible due to incompatible RPM PROFILE.

Use CALC:MARK:POS command to move marker on ORBIT Data and NYQUIST or POLAR Trace Coord.

Value entered cannot be zero.

Value entered must have same sign as current marker value.

WATERFALL STEPS must be > 1.

WATERFALL cannot be turned ON with FRNT/BACK display formats.

Waterfall markers require WATERFALL ON.

Waterfall operation not available with measurement running.

When the 102.4 kHz span is being used the antialias filter for channel 1 cannot be turned on.

X-axis units cannot be changed for this data.

Y unit selection is meaningless for the current combination of Meas Data and Trace Coord selections.

Y-axis scale matching not possible with current Trace Coord selections.

dB REF SETUP requires Trace Coordinate to be dB MAGNITUDE and XDCR UNIT to be Pascals, USER, or OFF.

dB SPL choice requires Trace Coordinate to be dB MAGNITUDE and XDCR UNIT to be Pascals.

dbm choice requires Trace Coordinate to be dB MAGNITUDE and XDCR UNIT to be OFF or V.

#### **-211: Trigger ignored.**

Bus trigger ignored when trigger type is not GPIB trigger.

Trigger received when not waiting for trigger.

**-212: Arm ignored.**

Arm ignored when arm type is not MANUAL ARM.  
Arm received when not waiting for arm.

**-220: Parameter error.**

Invalid Instrument State Parameter.

**-221: Settings conflict.**

Capture does not contain tach data.  
Capture frequency is not compatible with this measurement.  
Data edit invalid for Display Format WATERFALL.  
Invalid program state change requested.  
LogX not valid for negative X axis.  
Marker function result not available.  
Meas Data selection invalid for Display Format WATERFALL.  
Measurement mode incompatible with command.  
No TRIGGER With SWEPT SINE Instrument Mode.  
No WATERFALL display with DATA REGISTER, ORBIT, or CAPTURE data.  
No WATERFALL display with ORBIT Meas Data or NYQUIST or POLAR Trace Coord .  
No WATERFALL display with ORDER ANALYSIS Instrument Mode and TRACK ON.  
No WATERFALL display with ORDER TRACK data.  
No WATERFALL display with SWEPT SINE Instrument Mode.  
No WINDOW With HISTOGRAM Instrument Mode.  
No WINDOW With OCTAVE Instrument Mode.  
No WINDOW With SWEPT SINE Instrument Mode.  
SOURCE LEVEL is 0 Volts, measurement paused.  
Synthesis table must be pole-zero.  
Zoom capture data cannot be used with this measurement.

**-222: Data out of range.**

Data out of range.  
Out of range: value not changed.

**-224: Illegal parameter value.**

REAL format length is only 32 and 64.

**-230: Data corrupt or stale.**

Data does not contain SDF header information.

**-240: Hardware error**

Hardware error (see Fault Log).

**-241: Configuration error**

OPTION AY6 Add 2 Input Channels not installed.

**-250: Mass storage error.**

Bad mass storage parameter.  
Bad or unformatted disk.

## Error Messages

### Execution Errors

Can't name split file. Enter a shorter filename.  
Directories exist only on DOS file system.  
Directory must be empty before deletion.  
Disk file/unit possibly corrupt.  
External disk is NOT SS/80 protocol.  
External disk not responding.  
FORMAT aborted: file(s) are open.  
File does not contain MATH definitions.  
File does not contain LIMIT definitions.  
File does not contain a CAPTURE.  
File does not contain a STATE.  
File does not contain a TRACE.  
File does not contain a WATERFALL.  
File does not contain a data table.  
File system error.  
GPIB system controller needed.  
INSTALL aborted - invalid option.  
Improper file name.  
Improper file type.  
Improper mass storage unit specifier.  
Invalid SDF file format.  
Mass storage units must be the same when renaming.  
No data in the limit file to save.  
Operation failed on one (or more) files.  
Permission denied.  
SDF feature NOT supported.  
Source and destination units are same.  
Too many disk units active.  
Unexpected end of file.  
Wildcard expands to more than one file.  
Wildcard not allowed.

#### **-251: Missing mass storage.**

Mass storage unit not present.

#### **-252: Missing media.**

Disk not in drive

#### **-253: Corrupt media.**

Not a valid directory.

#### **-254: Media full.**

File too large, Press CONTINUE SAVE to split file.  
Insert next disk with file '[filename]', press CONTINUE RECALL.  
Insert next disk, press CONTINUE SAVE.  
Insufficient disk space. Press any key to continue.

**-255: Directory full.**

Full directory Press any key to continue.

**-256: File name not found.**

File name is undefined.

**-257: File name error.**

Duplicate file name.

**-258: Media protected.**

Write protected disk.

**-280: Program error.**

Instrument BASIC execution error. Refer to Instrument BASIC Users Handbook, Appendix A, "Error Messages."

**-283: Illegal variable name.**

**-284: Program currently running.**

**-285: Program syntax error.**

Downloaded program line must have a line number.  
ERROR 949 Syntax error at cursor.

## Device-Specific Errors

**-310: System error.**

Calibration Failure. Change state of AUTO CAL to remove this message. Key Path: [System Utility] - [CALIBRATN].  
Serial number already set.  
Uncalibrated data.

**-311: Memory error.**

EEPROM not initialized correctly.  
Out Of Memory.  
Out of Memory. Need [specified number of] bytes. See MEMORY USAGE.

**-315: Configuration memory lost.**

Save system configuration to EEPROM failed.



## Query Errors

**-400: Query error.**

[GPIB command which generated error].

**-410: Query interrupted.**

**-420: Query unterminated.**

**-430: Query deadlocked.**

**-440: Query unterminated after indefinite response.**

**-450: Query not allowed.**



## Instrument Modes

---



## Instrument Modes

This table indicates GPIB commands that are valid in each instrument mode. GPIB commands that cannot be executed in a particular instrument mode are listed as “not valid.”

Restrictions for use of these commands within the specific instrument mode may apply.

Table D-1. Valid GPIB Commands For Each Instrument Mode

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
*CAL?						
*CLS						
*ESE						
*ESR?						
*IDN?						
*OPC						
*OPC?						
*OPT?						
*PCB						
*PSC						
*RST						
*SRE						
*STB?						
*TRG				not valid		
*TST?						
*WAI						
ABORt						
ARM:IMMediate				not valid		
ARM:RPM:INCRement				not valid		
ARM:RPM:MODE				not valid		
ARM:RPM:THReshold	not valid			not valid	not valid	
ARM:SOURce				not valid		
ARM:TIMer				not valid		
CALCulate:ACTive						
CALCulate:CFIT:ABORt						
CALCulate:CFIT:COPI						
CALCulate:CFIT:DATA						

Instrument Modes

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
CALCulate:CFIT:DESTination						
CALCulate:CFIT:FREQuency[:AUTO						
CALCulate:CFIT:FREQuency:STARt						
CALCulate:CFIT:FREQuency:STOP						
CALCulate:CFIT:FSCale						
CALCulate:CFIT[:IMMEDIATE						
CALCulate:CFIT:ORDeR:AUTO						
CALCulate:CFIT:ORDeR:POLes						
CALCulate:CFIT:ORDeR:ZERos						
CALCulate:CFIT:TDElay						
CALCulate:CFIT:WEIGHt:AUTO						
CALCulate:CFIT:WEIGHt:REGister						
CALCulate:DATA?						
CALCulate:DATA:HEADer:POINTs?						
CALCulate:FEED						
CALCulate:FORMat						
CALCulate:GDAPerture:APERture		not valid	not valid		not valid	not valid
CALCulate:LIMit:BEEP[:STATe						
CALCulate:LIMit:FAIL?						
CALCulate:LIMit:LOWer:CLEar:IMMEDIATE						
CALCulate:LIMit:LOWer:MOVE:Y						
CALCulate:LIMit:LOWer:REPort[:DATA?						
CALCulate:LIMit:LOWer:REPort:YDATA?						
CALCulate:LIMit:LOWer:SEGment						
CALCulate:LIMit:LOWer:SEGment:CLEar						
CALCulate:LIMit:LOWer:TRACe[:IMMEDIATE						
CALCulate:LIMit:STATe						
CALCulate:LIMit:UPPer:CLEar:IMMEDIATE						
CALCulate:LIMit:UPPer:MOVE:Y						
CALCulate:LIMit:UPPer:REPort:DATA?						
CALCulate:LIMit:UPPer:REPort:YDATA?						
CALCulate:LIMit:UPPer:SEGment						
CALCulate:LIMit:UPPer:SEGment:CLEar						
CALCulate:LIMit:UPPer:TRACe:IMMEDIATE						
CALCulate:MARKer:BAND:STARt						
CALCulate:MARKer:BAND:STOP						
CALCulate:MARKer:COUPlEd:STATe						
CALCulate:MARKer:DTABle:CLEar:IMMEDIATE						
CALCulate:MARKer:DTABle:COPI						
CALCulate:MARKer:DTABle:DATA						
CALCulate:MARKer:DTABle:X:DATA						

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
CALCulate:MARKer:DTABLE:X:DELeTe						
CALCulate:MARKer:DTABLE:X:INSert						
CALCulate:MARKer:DTABLE:X:LABel						
CALCulate:MARKer:DTABLE:X:SElect:POINt						
CALCulate:MARKer:FUNcTion						
CALCulate:MARKer:FUNcTion:RESult?						
CALCulate:MARKer:HARMonic:COUNt						
CALCulate:MARKer:HARMonic:FUNDamental						
CALCulate:MARKer:MAXimum:GLOBal						
CALCulate:MARKer:MAXimum:GLOB:TRACk						
CALCulate:MARKer:MAXimum:LEFT						
CALCulate:MARKer:MAXimum:RIGHT						
CALCulate:MARKer:MODE						
CALCulate:MARKer:POSition						
CALCulate:MARKer:POSition:POINt						
CALCulate:MARKer:REFerence:X						
CALCulate:MARKer:REFerence:Y						
CALCulate:MARKer:SIDeband:CARRier						
CALCulate:MARKer:SIDeband:COUNt						
CALCulate:MARKer:SIDeband:INCReMent						
CALCulate:MARKer:STATe						
CALCulate:MARKer:X:ABSolute						
CALCulate:MARKer:X:RELative						
CALCulate:MARKer:Y:ABSolute?						
CALCulate:MARKer:Y:RELative						
CALCulate:MATH:CONStant						
CALCulate:MATH:DATA						
CALCulate:MATH:EXPRession						
CALCulate:MATH:SElect						
CALCulate:MATH:STATe						
CALCulate:SYNThesis:COpy						
CALCulate:SYNThesis:DATA						
CALCulate:SYNThesis:DEStination						
CALCulate:SYNThesis:FSCale						
CALCulate:SYNThesis:GAIN						
CALCulate:SYNThesis:IMMediate						
CALCulate:SYNThesis:SPACing						
CALCulate:SYNThesis:TDElay						
CALCulate:SYNThesis:TTPe						
CALCulate:UNIT:AMPLitude						
CALCulate:UNIT:ANGLe						

Instrument Modes

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
CALCulate:UNIT:DBReference						
CALCulate:UNIT:DBReference:IMPedance						
CALCulate:UNIT:DBReference:USER:LABel						
CALC:UNIT:DBReference:USER:REFerence						
CALCulate:UNIT:MECHanical						
CALCulate:UNIT:VOLTage						
CALCulate:UNIT:X		not valid	not valid			time data only
CALCulate:UNIT:X:ORDer:FACTor		not valid	not valid			time data only
CALCulate:UNIT:X:USER:FREQuency:FACTor		not valid	not valid			time data only
CALCulate:UNIT:X:USER:FREQuency:LABel		not valid	not valid			time data only
CALCulate:UNIT:X:USER:TIME:FACTor		not valid	not valid			time data only
CALCulate:UNIT:X:USER:TIME:LABel		not valid	not valid			time data only
CALCulate:WATerfall:COUNT				not valid		
CALCulate:WATerfall:DATA						
CALCulate:WATerfall:SLICe:COPY				not valid		
CALCulate:WATerfall:SLICe:SElect				not valid		
CALCulate:WATerfall:SLICe:SElect:POINt				not valid		
CALCulate:WATerfall:TRACe:COPY				not valid		
CALCulate:WATerfall:TRACe:SElect				not valid		
CALCulate:WATerfall:TRACe:SElect:POINt				not valid		
CALCulate:X:DATA						
CALibration:ALL?						
CALibration:AUTO						
DISPlay:ANNOtation:ALL						
DISPlay:BODE		not valid	not valid		not valid	not valid
DISPlay:BRIGHtness						
DISPlay:ERRor						
DISPlay:EXTeRnal:STATe						
DISPlay:FORMat						
DISPlay:GPIB:ECHO						
DISPlay:PROGram:KEY:BOX						
DISPlay:PROGram:KEY:BRACket						
DISPlay:PROGram:MODE						
DISPlay:PROGram:VECTor:BUFFer:STATe						
DISPlay:RPM:STATe						
DISPlay:STATe						



GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
DISPlay:TCAPture:ENVELOpe:STATe						
DISPlay:VIEW						
DISPlay:WINDow:DTABLE:MARKer:STATe						
DISPlay:WINDow:DTABLE:STATe						
DISPlay:WINDow:LIMit:STATe						
DISPlay:WINDow:POLar:CLOCKwise						
DISPlay:WINDow:POLar:ROTation						
DISPlay:WINDow:TRACe:APOWer:STATe						
DISPlay:WINDow:TRACe:BPOWer:STATe						
DISPlay:WINDow:TRACe:GRAT:GRID:STATe						
DISPlay:WINDow:TRACe:LABel						
DISPlay:WINDow:TRAC:LABel:DEF:STATe						
DISPlay:WINDow:TRACe:X:MATch						
DISPlay:WINDow:TRACe:X:SCALe:AUTO						
DISPlay:WINDow:TRACe:X:SCALe:LEFT						
DISPlay:WINDow:TRACe:X:SCALe:RIGHT						
DISPlay:WINDow:TRACe:X:SPACing						
DISPlay:WINDow:TRACe:Y:MATch						
DISPlay:WINDow:TRACe:Y:SCALe:AUTO						
DISPlay:WINDow:TRAC:Y:SCAL:BOTTom						
DISPlay:WINDow:TRACe:Y:SCALe:CENTer						
DISPlay:WINDow:TRAC:Y:SCAL:PDIVision						
DISPlay:WINDow:TRAC:Y:SCAL:REFerence						
DISPlay:WINDow:TRACe:Y:SCALe:TOP						
DISPlay:WINDow:TRACe:Y:SPACing						
DISPlay:WINDow:WATerfall:BASEline				not valid		
DISPlay:WINDow:WATerfall:COUNt				not valid		
DISPlay:WINDow:WATerfall:HEIGht				not valid		
DISPlay:WINDow:WATerfall:HIDDen				not valid		
DISPlay:WINDow:WATerfall:SKEW				not valid		
DISPlay:WINDow:WATerfall:SKEW:ANGLE				not valid		
FORMat:DATA						
HCOPy:COLor:DEFaultn						
HCOPy:DESTination						
HCOPy:DEVIce:LANGUage						
HCOPy:DEVIce:SPEEd						
HCOPy:IMMediate						
HCOPy:ITEM:ALL:IMMediate						
HCOPy:ITEM:FFEEd:STATe						
HCOPy:ITEM:LABel:COLor						
HCOPy:ITEM:LABel:STATe						

Instrument Modes

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
HCOPY:ITEM:LABel:TEXT						
HCOPY:ITEM:TDSTamp:FORMat						
HCOPY:ITEM:TDSTamp:STATe						
HCOPY:ITEM:WINDow:TRACe:COLor						
HCOPY:ITEM:WINDow:TRACe:GRAT:COLor						
HCOP:ITEM:WINDow:TRACe:GRAT:IMMediate						
HCOPY:ITEM:WINDow:TRACe:IMMediate						
HCOPY:ITEM:WINDow:TRACe:LIMit:LTYPe						
HCOPY:ITEM:WINDow:TRACe:LTYPe						
HCOPY:ITEM:WINDow:TRACe:MARKer:COLor						
HCOP:ITEM:WIND:TRACe:MARKer:IMMediate						
HCOP:ITEM:WIND:TRAC:MARK:REF:IMM						
HCOPY:PAGE:DIMensions:AUTO						
HCOPY:PAGE:DIMensions:USER:LLEFt						
HCOPY:PAGE:DIMensions:USER:URIGHt						
HCOPY:PLOT:ADDRes						
HCOPY:PRINt:ADDRes						
HCOPY:TITle						
INITiate:CONTinuous						
INITiate:IMMediate						
INPut:BIAS:STATe						
INPut:COUPling						
INPut:FILTer:AWEighting:STATe						not valid
INPut:FILTer:LPASs:STATe						not valid
INPut:LOW						
INPut:REFerence:DIRection						
INPut:REFerence:POINt						
INPut:STATe						
INSTrument:NSElect						
INSTrument:SElect						
MEMory:CATalog:ALL						
MEMory:CATalog:NAME						
MEMory:DELeTe:ALL						
MEMory:DELeTe:NAME						
MEMory:FREE:ALL						
MMEemory:COPI						
MMEemory:DELeTe						
MMEemory:DISK:ADDRes						
MMEemory:DISK:UNIT						
MMEemory:FSYSstem						
MMEemory:INITialize						

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
MMEemory:LOAD:CFIT						
MMEemory:LOAD:CONTinue						
MMEemory:LOAD:DTABLE:TRACe						
MMEemory:LOAD:LIMit:LOWer:TRACe						
MMEemory:LOAD:LIMit:UPPer:TRACe-						
MMEemory:LOAD:MATH						
MMEemory:LOAD:PROGram						
MMEemory:LOAD:STATe						
MMEemory:LOAD:SYNTHeSis						
MMEemory:LOAD:TCAPture						
MMEemory:LOAD:TRACe						
MMEemory:LOAD:WATerfall						
MMEemory:MOVE						
MMEemory:MSIS						
MMEemory:STORE:CFIT						
MMEemory:STORE:CONTInue						
MMEemory:STORE:DTABLE:TRACe						
MMEemory:STORE:LIMit:LOWer:TRACe						
MMEemory:STORE:LIMit:UPPer:TRACe						
MMEemory:STORE:MATH						
MMEemory:STORE:PROGram						
MMEemory:STORE:STATe						
MMEemory:STORE:SYNTHeSis						
MMEemory:STORE:TCAPture						
MMEemory:STORE:TRACe						
MMEemory:STORE:WATerfall						
OUTPut:STATe						
PROGram:EDIT:ENABle						
PROGram:EXPLicit:DEFine						
PROGram:EXPLicit:LABel						
PROGram:SElected:DEFine						
PROGram:SElected:DELeTe:ALL						
PROGram:SElected:DELeTe:SElected						
PROGram:SElected:LABel						
PROGram:SElected:MALLocate						
PROGram:SElected:NAME						
PROGram:SElected:NUMBer						
PROGram:SElected:STATe						
PROGram:SElected:STRing						
SENSe:AVERage:CONFidence	not valid			not valid	not valid	not valid
SENSe:AVERage:COUNt				not valid		not valid

Instrument Modes

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
SENSe:AVERage:HOLD	not valid			not valid	not valid	not valid
SENSe:AVERage:IMPulse	not valid			not valid	not valid	not valid
SENSe:AVERage:IREsult:RATE				not valid		not valid
SENSe:AVERage:IREsult:STATe						
SENSe:AVERage:PREView		not valid	not valid	not valid	not valid	not valid
SENSe:AVERage:PREView:ACcept		not valid	not valid	not valid	not valid	not valid
SENSe:AVERage:PREView:REJect		not valid	not valid	not valid	not valid	not valid
SENSe:AVERage:PREView:TIME		not valid	not valid	not valid	not valid	not valid
SENSe:AVERage:STATe				not valid		not valid
SENSe:AVERage:TCONtrol				not valid		
SENSe:AVERage:TIME	not valid			not valid	not valid	
SENSe:AVERage:TYPE				not valid		not valid
SENSe:FEED				not valid		
SENSe:FREQuency:BLOCKsize		not valid	not valid	not valid		not valid
SENSe:FREQuency:CENTer		not valid	not valid		not valid	not valid
SENSe:FREQuency:MANual	not valid	not valid	not valid		not valid	not valid
SENSe:FREQuency:RESolution		not valid	not valid			not valid
SENSe:FREQuency:RESolution:AUTO	not valid	not valid	not valid		not valid	not valid
SENSe:FREQuency:RES:AUTO:MCHange	not valid	not valid	not valid		not valid	not valid
SENSe:FREQuency:RES:AUTO:MINimum	not valid	not valid	not valid		not valid	not valid
SENSe:FREQuency:RESolution:OCTave	not valid		not valid	not valid	not valid	not valid
SENSe:FREQuency:SPAN		not valid	not valid		not valid	
SENSe:FREQuency:SPAN:FULL		not valid	not valid	not valid		not valid
SENSe:FREQuency:SPAN:LINK		not valid	not valid		not valid	not valid
SENSe:FREQuency:START			not valid		not valid	not valid
SENSe:FREQuency:STEP:INCRement		not valid	not valid		not valid	not valid
SENSe:FREQuency:STOP			not valid		not valid	not valid
SENSe:HISTogram:BINs	not valid	not valid	not valid	not valid	not valid	
SENSe:ORDer:MAXimum	not valid	not valid		not valid	not valid	not valid
SENSe:ORDer:RESolution	not valid	not valid		not valid	not valid	not valid
SENSe:ORDer:RESolution:TRACk	not valid	not valid		not valid	not valid	not valid
SENSe:ORDer:RPM:MAXimum	not valid	not valid		not valid	not valid	not valid
SENSe:ORDer:RPM:MINimum	not valid	not valid		not valid	not valid	not valid
SENSe:ORDer:TRACk	not valid	not valid		not valid	not valid	
SENSe:REFerence			not valid			not valid
SENSe:REJect:STATe		not valid	not valid	not valid		not valid
SENSe:SWEEp:DIRection	not valid	not valid	not valid		not valid	not valid
SENSe:SWEEp:DWELI	not valid	not valid	not valid		not valid	not valid
SENSe:SWEEp:MODE	not valid	not valid	not valid		not valid	not valid
SENSe:SWEEp:OVERlap		not valid	not valid	not valid		not valid
SENSe:SWEEp:SPACing	not valid	not valid	not valid		not valid	not valid

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
SENSe:SWEEp:STIME	not valid	not valid	not valid		not valid	not valid
SENSe:SWEEp:TIME		not valid	not valid	not valid		
SENSe:TCAPture:ABORt				not valid		
SENSe:TCAPture:DELeTe						
SENSe:TCAPture:IMMEdiate				not valid		
SENSe:TCAPture:LENGth						
SENSe:TCAPture:MALLocate						
SENSe:TCAPture:STARt						
SENSe:TCAPture:STOP						
SENSe:TCAPture:TACHometer:RPM:MAX						
SENSe:TCAPture:TACHometer:STATe						
SENSe:VOLTage:DC:RANGe:AUTO						
SENSe:VOLT:DC:RANGe:AUTO:DIRectioN			not valid			
SENSe:VOLT:DC:RANGe:UNIT:MODE						
SENSe:VOLT:DC:RANGe:UNIT:USER:LABel						
SENSe:VOLT:DC:RANG:UNIT:USER:SFACtor						
SENSe:VOLT:DC:RANG:UNIT:USER:STATe						
SENS:VOLT:DC:RANG:UNIT:USER:UNIT:XDCR						
SENSe:VOLTage:DC:RANGe:UPPer						
SENSe:WINDow:EXPOntial		not valid	not valid	not valid	not valid	not valid
SENSe:WINDow:FORCe		not valid	not valid	not valid	not valid	not valid
SENSe:WINDow:ORDeR:DC	not valid	not valid		not valid	not valid	not valid
SENSe:WINDow:TYPE		not valid		not valid		not valid
SOURce:BURSt		not valid	not valid	not valid		
SOURce:FREQuency:CW				not valid		
SOURce:FREQuency:FIXed				not valid		
SOURce:FUNCTioN:SHAPE				not valid		
SOURce:USER:REGister		not valid	not valid	not valid		
SOURce:USER:REPeat		not valid	not valid	not valid		
SOURce:VOLTage:LEVel:AUTO	not valid	not valid	not valid		not valid	not valid
SOUR:VOLT:LEV:IMMEdiate:AMPLitude						
SOUR:VOLT:LEV:IMMEdiate:AMPLitude:OFFset						
SOURce:VOLTage:LEVel:REFerence	not valid	not valid	not valid		not valid	not valid
SOUR:VOLTage:LEVel:REFerence:CHANnel	not valid	not valid	not valid		not valid	not valid
SOUR:VOLT:LEVel:REFerence:TOLerance	not valid	not valid	not valid		not valid	not valid
SOURce:VOLTage:LIMit:AMPLitude	not valid	not valid	not valid		not valid	not valid
SOURce:VOLTage:LIMit:INPut	not valid	not valid	not valid		not valid	not valid
SOURce:VOLTage:SLEW	not valid	not valid	not valid		not valid	not valid
STATus:DEVice:CONDition?						
STATus:DEVice:ENABLE						
STATus:DEVice:EVENt?						

Instrument Modes

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
STATus:DEvice:NTRansition						
STATus:DEvice:PTRansition						
STATus:OPERation:CONDition?						
STATus:OPERation:ENABle						
STATus:OPERation:EVENT?						
STATus:OPERation:NTRansition						
STATus:OPERation:PTRansition						
STATus:PRESet						
STATus:QUEStionable:CONDition?						
STATus:QUEStionable:ENABle						
STATus:QUEStionable:EVENT?						
STATus:QUEStionable:LIMit:CONDition?						
STATus:QUEStionable:LIMit:ENABle						
STATus:QUEStionable:LIMit:EVENT?						
STATus:QUEStionable:LIMit:NTRansition						
STATus:QUEStionable:LIMit:PTRansition						
STATus:QUEStionable:NTRansition						
STATus:QUEStionable:PTRansition						
STATus:QUEStionable:VOLTage:CONDition?						
STATus:QUEStionable:VOLTage:ENABle						
STATus:QUEStionable:VOLTage:EVENT?						
STATus:QUEStionable:VOLTage:NTRansition						
STATus:QUEStionable:VOLTage:PTRansition						
STATus:USER:ENABle						
STATus:USER:EVENT?						
STATus:USER:PULSe-?						
SYSTem:BEEPer:IMMEDIATE						
SYSTem:BEEPer:STATe						
SYSTem:COMMunicate:GPIB:SELF:ADDRes						
SYST:COMMunicate:SERial:RECeive:BAUD						
SYSTem:COMMunicate:SERial:RECeive:BITS						
SYST:COMMunicate:SERial:RECeive:PACe						
SYST:COM:SERial:RECeive:PARity:CHECK						
SYST:COMM:SERial:RECeive:PARity:TYPE						
SYST:COMM:SERial:RECeive:SBITs						
SYST:COMM:SERial:TRANmit:PACe						
SYSTem:DATE						
SYSTem:ERRor						
SYSTem:FAN:STATe						
SYSTem:FLOG:CLEAr						
SYSTem:KEY						

GPIB Command	Instrument Mode (INST:SEL )					
	FFT	OCT	ORD	SINE	CORR	HIST
SYSTem:KLOCK						
SYSTem:POWer:SOUce						
SYSTem:POWer:STATe						
SYSTem:PRESet						
SYSTem:SET						
SYSTem:TIME						
SYSTem:VERSion						
TEST:LOG:CLEar						
TEST:LONG						
TEST:LONG:RESult?						
TRACe:DATA						
TRACe:WATERfall:DATA						
TRACe:X:DATA						
TRACe:X:UNIT						
TRACe:Z:DATA						
TRACe:Z:UNIT						
TRIGger:EXTernal:FILTer:LPASs:STATe				not valid		
TRIGger:EXTernal:LEVel				not valid		
TRIGger:EXTernal:RANGe					not valid	
TRIGger:IMMEDIATE				not valid		
TRIGger:LEVel		not valid	not valid	not valid		
TRIGger:SLOPe		not valid	not valid	not valid		
TRIGger:SOURce				not valid		
TRIGger:STARt			not valid	not valid		
TRIGger:TACHometer:HOLDoff				not valid		
TRIGger:TACHometer:LEVel				not valid		
TRIGger:TACHometer:PCOunt				not valid		
TRIGger:TACHometer:RANGe				not valid		
TRIGger:TACHometer:RPM?				not valid		
TRIGger:TACHometer:SLOPe				not valid		





## Determining Units

---



## Determining Units

---

The following tables (table E-1 - table E-8) show the units available for the Y-axis. The tables indicate which Y-axis units are available for each measurement data selection and which GPIB commands result in setting the Y-axis units. See the “Command Reference” in this guide for a description of these commands.

Table E-1 specifies the default Y-axis unit for each measurement data type (specified with the CALC:FEED command) and each trace coordinate system (specified with the CALC:FORMat command).

You can change the Y-axis unit for some types of measurement data with the CALC:UNIT:VOLT and CALC:UNIT:AMPLitude commands. See table E-2 for vertical unit selection and tables E-3 — E-8 for amplitude selection.

---

### Note

If a measurement data type is not listed in table E-2, the default unit listed in table E-1 is always used. You cannot select the Y-axis (vertical) units in these cases. Tables E-3 - E-8 only apply to the measurement data listed in table E-2.

---

A dB magnitude reference level may be applied to traces with dB magnitude coordinates. The dB reference scaling is applied after transducer units have been applied. See the CALCulate:UNIT:DBRefence commands for more information.

## Determining Units

The analyzer determines the default Y-axis unit based upon the specified measurement data and trace coordinate system. Table E-1 lists these default Y-axis units.

Units for data registers (D[1|2 . . . |8]) and waterfall registers (W[1|2| . . . |8]) are dependent upon the type of measurement data stored in the register. See the appropriate measurement data row for valid unit selections.

Table E-1. Default Y-axis Units for Measurement Data and Trace Coordinates.

Measurement Data	Specified Trace Coordinates			
Type of Measurement CALC:FEED?	Linear or Logarithmic Magnitude CALC:FORM MLIN	dB Magnitude CALC:FORM MLOG	Phase/Unwrapped Phase CALC:FORM [PHAS   UPH] *	Real/Imaginary CALC:FORM [REAL   IMAG]
auto correlation CALC:FEED 'XTIM:CORR'	VPK2	DBV	DEG	VPK2
capture buffer CALC:FEED 'TCAP'	VPK	DBV	DEG	VPK
Coherence CALC:FEED 'XFR:POW:COH'	none	none	DEG	none
composite power CALC:FEED 'XFR:POW:COMP'	VRMS	DBVRMS	DEG	VRMS
cross correlation CALC:FEED 'XTIM:CORR:CROS'	VPK2	DBV	DEG	VPK2
cross spectrum ** CALC:FEED 'XFR:POW:CROS'	VRMS2	DBVRMS	DEG	VRMS2
cumulative density function CALC:FEED 'XTIM:VOLT:CDF'	none	DB	DEG	none
frequency response CALC:FEED 'XFR:POW:RAT'	none	DB	DEG	none
Histogram CALC:FEED 'XTIM:VOLT:HIST'	COUNT	DBCOUNT	DEG	COUNT
linear spectrum CALC:FEED 'XFR:POW:LIN'	VRMS	DBVRMS	DEG	VRMS
normalized variance CALC:FEED 'XFR:POW:VAR'	none	DB	DEG	none
Orbits CALC:FEED 'XVOL:VOLT'	not valid	not valid	not valid	VPK
order track CALC:FEED 'XORD:TRACK'	VRMS	DBVRMS	DEG	VRMS
power spectrum CALC:FEED 'XFR:POW'	VRMS2	DBVRMS	DEG	VRMS2
probability density function CALC:FEED 'XTIM:VOLT:PDF'	1/V	DB1/V	DEG	1/V
rpm profile CALC:FEED 'XRPM:PROF'	COUNT***	DBCOUNT	DEG	COUNT
Time CALC:FEED 'XTIM:VOLT'	VPK	DBV	DEG	VPK

Measurement Data	Specified Trace Coordinates			
Type of Measurement CALC:FEED?	Linear or Logarithmic Magnitude CALC:FORM MLIN	dB Magnitude CALC:FORM MLOG	Phase/Unwrapped Phase CALC:FORM [PHAS   UPH] *	Real/Imaginary CALC:FORM [REAL   IMAG]
windowed time CALC:FEED 'XTIM:VOLT:WIND'	VPK	DBV	DEG	VPK

\* This is the default unit. You can select radians (RAD).

\*\* This is the default unit. You can specify VPK using the CALC:UNIT AMPL command.

\*\*\* This is the default unit. The unit is S (seconds) if order track is on (SENSe:ORDeR:TRACk:STATe ON).

## Determining Units

For linear spectrum and power spectrum measurement data, the Y-axis vertical unit is set with the CALCulate:UNIT:VOLTage command. See table E-2 for valid Y-axis vertical units. An “X” indicates a valid selection.

Table E-2. Valid Unit Selections for CALC:UNIT:VOLT

Measurement Data CALC:FEED command (INST:SEL command)	CALC:UNIT:VOLT				
	V	V2	V/RTHZ	V2/HZ	V2S/HZ
Composite Power CALC:FEED 'XFR:POW:COMP' (INST:SEL ORD)	X	X			
Linear Spectrum CALC:FEED 'XFR:POW:LIN' (INST:SEL FFT)	X	X	X	X	X
Linear Spectrum CALC:FEED 'XFR:POW:LIN' (INST:SEL SINE)	X	X			
Order Track CALC:FEED 'XORD:TRACK' (INST:SEL ORD)	X	X			
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL FFT)	X	X	X	X	X
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL OCT)	X	X	X	X	X
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL ORD)	X	X			
Power Spectrum CALC:FEED 'XFR:POW' (INST:SEL ORD)	X	X			

Units for data registers (D[1|2 . . . |8]) and waterfall registers (W[1|2| . . . |8]) are dependent upon the type of measurement data stored in the register. See the appropriate measurement data row for valid unit selections.

The CALCulate:UNIT:VOLTage command is only valid for the measurement data listed in the table. It is not valid for any other types of measurement data. If you do not know the current measurement data selection, send the query, CALCulate[1|2]:FEED?.

For linear spectrum and power spectrum measurement data, the unit of amplitude for the Y-axis scale is set with the CALCulate:UNIT:AMPLitude command. In addition, tables E-6 -E-8 show the unit of amplitude with transducer units (enabled with the [SENSe:]VOLTage[:DC]:RANGe:UNIT:USER[:STATe] command).

Tables E-3 - E-8 only apply to the measurement data listed in table E-2.

Table E-3. Y-axis Units when CALC:UNIT:AMPL PEAK and VOLT:RANG:UNIT:USER OFF.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	V VPK	DBV
V2	V2	DBV
V/RTHZ	V/RTHZ	DBV/RTHZ
V2/HZ	V2/HZ	DBV/RTHZ
V2S/HZ	V2S/HZ	DBV2S/HZ

Table E-4. Y-axis Units when CALC:UNIT:AMPL PP and VOLT:RANG:UNIT:USER OFF.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	VPP	DBVPP
V2	VPP2	DBVPP
V/RTHZ	VPP/RTHZ	DBVPP/RTHZ
V2/HZ	VPP2/HZ	DBVPP/RTHZ
V2S/HZ	VPP2S/HZ	DBVPP2S/HZ

Table E-5. Y-axis Units when CALC:UNIT:AMPL RMS and VOLT:RANG:UNIT:USER OFF.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	VRMS	DBVRMS
V2	VRMS2	DBVRMS
V/RTHZ	VRMS/RTHZ	DBVRMS/RTHZ
V2/HZ	VRMS2/HZ	DBVRMS/RTHZ
V2S/HZ	VRMS2S/HZ	DBVRMS2S/HZ

Table E-6. Y-axis Units when CALC:UNIT:AMPL PEAK and VOLT:RANG:UNIT:USER ON.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	EU	DBEU
V2	EU2	DBEU
V/RTHZ	EU/RTHZ	DBEU/RTHZ
V2/HZ	EU2/HZ	DBEU/RTHZ
V2S/HZ	EU2S/HZ	DBEU2S/HZ

Table E-7. Y-axis Units when CALC:UNIT:AMPL PP and VOLT:RANG:UNIT:USER ON.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	EUPP	DBEUPP
V2	EUPP2	DBEUPP
V/RTHZ	EUPP/RTHZ	DBEUPP/RTHZ
V2/HZ	EUPP2/HZ	DBEUPP/RTHZ
V2S/HZ	EUPP2S/HZ	DBEUPP2S/HZ

Table E-8. Y-axis Units when CALC:UNIT:AMPL RMS and VOLT:RANG:UNIT:USER ON.

CALC:UNIT:VOLT	Trace Coordinate Setting	
	CALC:FORM [MLIN   REAL   IMAG]	CALC:FORM MLOG
V	EURMS	DBEURMS
V2	EURMS2	DBEURMS
V/RTHZ	EURMS/RTHZ	DBEURMS/RTHZ
V2/HZ	EURMS2/HZ	DBEURMS/RTHZ
V2S/HZ	EURMS2S/HZ	DBEURMS2S/HZ



## Example Programs

---



## Example Programs

---

These listings of example programs written for the Agilent 35670A demonstrate many important programming concepts, specifically, transferring data.

The programs are written in BASIC for use on an HP Series 300 computer, on a PC with Instrument BASIC for Windows, or on the Agilent 35670A with Instrument BASIC (Option 1C2). They contain numerous comments to make them easily adaptable to other languages or programs. The example programs are on the Agilent 35670A GPIB Example Programs disk.

The listings, which are organized alphabetically by name, are as follows:

### ARBSRC

Loads one of the Agilent 35670A's data registers with data generated on a computer. You can view the results if you connect the Agilent 35670A's arbitrary source (Option 1D4) to Channel 1.

### COPYDATA

Copies measurement data to a data register.

### FREQRESP

Sets up a frequency response measurement, runs the measurement and then transfers the data.

### GETASCII

Reads trace data values in ASCII and displays the first ten values on the display.

### GETBIN

Queries power spectrum data from Channel 1 in binary and displays the first ten values on the display.

### GETCAP

Reads time-capture data from the analyzer, modifies the data, and writes the data back to the analyzer

### GETCURF

Queries curve fit data and transfers it to an external controller.

### GETWATR

Sets up and runs an octave measurement in a waterfall display. Transfers the frequency and amplitude values.

## Example Programs

### TIMEBIN

Queries Channel 1 time data in binary and displays the first ten values on the display.

### XFER

C-language program that uploads and downloads time-capture files between the analyzer and a PC.

For listings of example programs which focus on measurement synchronization, passing control, and generating service requests (SRQs), see the GPIB Programmer's Guide. For more information about programming with Instrument BASIC, see Using Instrument BASIC with the Agilent 35670A .

## ARBSRC

```

10   !  ARBSRC
20   !-----
30   !BASIC/IBASIC Program; ARB_EXAMPLE
40   ! This program sets up a register header, generates data of the correct
50   ! size and downloads it into the instrument. It then allows the user to
60   ! view the results, if the source is connected to Channel 1. This program
70   ! will also distinguish between IBASIC and BASIC run modes.
80   !-----
90   !
100  ON ERROR GOTO Not_ibasic !
110  Device=800
120  CLEAR Device
130  GOTO Start_prog
140  !
150 Not_ibasic:
160  !
170 Start_prog:
180  OFF ERROR
190  !
200  !
210  ASSIGN @Analyzer TO Device
220  !
230  !
240  ASSIGN @Analyzer_bin TO DEVICE;FORMAT OFF
250  !
260  OUTPUT @Analyzer;"*RST"
270  OUTPUT @Analyzer;"inp2 on"
280  OUTPUT @Analyzer;"ABOR;:INIT; *WAI"
290  OUTPUT @Analyzer;"INIT:CONT OFF"
300  !
310  INTEGER Res,Max_i,T
320  DIM Tones(1)
330  DIM D(20000)
340  RAD
350  !
360  Span=102400
370  Res=400
380  !
390  Tones(0)=7680
400  Tones(1)=32000
410  !

```

Program listing continues on page F-5.

Example Programs  
ARBSRC

## Comments for ARBSRC

- 100 - 180     Determines where the program is running; either in the analyzer or on an external controller.
- 110     Specify interface select code and primary address for Instrument BASIC.
- 170     Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 210     Allows device select code to be independent of address.
- 240     Assign an I/O path for binary transfers.
- 260     Reset the analyzer.
- 270     Put the analyzer in 1-channel mode.
- 280     Start the measurement.
- 290     Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 310     Reserve space for the variables Res, Max\_i, and T.
- 320     Reserve space for the array of tone values.
- 330     Reserve space for the array D to accommodate a large number of values. D is an array of time waveform data for the arbitrary source.
- 340     Select radians as unit of angular measure.
- 360     Set the variable, Span, to maximum.
- 370     Set the variable, Res, to 400 line resolution.
- 390 - 400     Specify the frequency of the example tones.

```

420  SELECT (Res)
430  CASE 100
440    Max_i=255
450  CASE 200
460    Max_i=511
470  CASE 400
480    Max_i=1023
490  CASE 800
500    Max_i=2047
510  CASE ELSE
520    PRINT "ILLEGAL MEASUREMENT RESOLUTION"
530    PAUSE
540  END SELECT
550  !
560  !
570  REDIM D(Max_i)
580  !
590  Dt=Res/Span/(Max_i+1)
600  FOR T=0 TO 1
610    FOR I=0 TO Max_i
620      D(I)=D(I)+SIN(2*PI*Tones(T)*I*Dt)
630    NEXT I
640  NEXT T
650  !
660  !
670  IF Device=800 THEN
680    OUTPUT @Analyzer;"DISP:FORM QUAD; PROG LOW"
690  ELSE
700    OUTPUT @Analyzer;"DISP:FORM ULOW;"
710  END IF
720  !
730  FOR I=0 TO 9
740    PRINT D(I)
750  NEXT I
760  !
770  !
780  OUTPUT @Analyzer;"FORM:DATA REAL, 64"
790  !
800  !
810  OUTPUT @Analyzer;"CALC1:FEED `XTIM:VOLT 1`;*WAI"
820  !
830  !
850  !
850  !
860  OUTPUT @Analyzer;"TRAC:DATA D1,TRAC1;*WAI"
870  !
880  !
890  OUTPUT @Analyzer;"TRAC:DATA D1,#0";
900  OUTPUT @Analyzer_bin;D(*)
910  OUTPUT @Analyzer;CHR$(10) END

```

Program listing continues on page F-7.

## Comments for ARBSRC (continued)

- 420 - 540    Verify that a legal resolution value has been selected. An error message appears if an illegal value is selected.
- 570    Allocate space for the tone data that will be loaded into the data register.
- 590    Compute the delta t (D) between time points in the tone data.
- 600 - 640    Compute the tone data for two tones.  $\text{SIN}(2*\text{PI}*\text{Tones}(T)*I*\text{D}5)$  generates a time waveform for a tone. The first time through the loop generates time data for the first tone. The second time through the loop, adds in time waveform data for the second tone.
- 670 - 710    Setup a display window.
- 670 - 680    If the program is running in Instrument BASIC on the analyzer, set the display format to QUAD. Print to the lower half of the display.
- 700    If the program is running on an external controller, set the display to UPPER/LOWER format.
- 730 - 750    Print the first ten values in data register D1.
- 780    The data block format (FORMAT:DATA) is 64-bit binary.
- 810    Display Channel 1 time data in trace A.
- 860    Store Channel 1 time data in data register D1. The data register contains real data because we saved baseband time data (start frequency = 0) in the data register.
- Before a data register can be loaded with new data from GPIB, the register must be initialized so it can accept the proper amount and representation of data. Use the CALC1:FEED command to set the trace data to the type of data you will be putting in the data register, then save the trace into the data register using the TRAC:DATA command.
- 890    TRAC:DATA D1,#0 uses the indefinite length block to load the data into data register D1. The # identifies the data as block data, the "0" after the # indicates the indefinite length block. the last byte of the output is a new line character sent with END.
- 900    Output the data in binary.
- 910    Output a line feed with EOI (new line) at the end of the data block.



```
920  !
930  !
940  !
950  OUTPUT @Analyzer;"SOUR:FUNC USER "
960  OUTPUT @Analyzer;"SOUR:USER D1"
970  OUTPUT @Analyzer;"SOUR:USER:REP ON"
980  OUTPUT @Analyzer;"SOUR:VOLT .1 VPK"
990  OUTPUT @Analyzer;"OUTP ON"
1000 !
1010 OUTPUT @Analyzer;"INIT"
1020 !
1030 END
```

Example Programs  
ARBSRC

Comments for ARBSRC (continued)

950 - 990    Setup the arbitrary source to output the signal stored in the data register.

950    Specify the arbitrary source.

960    Specify the source output from data register D1.

970    Set the source to output data continuously.

980    Set the source level.

990    Turn on the arbitrary source.

1010    Start the measurement.

## COPYDATA

```

10 ! COPYDATA
20 !-----
30 ! BASIC/IBASIC Programs; Copy Data
40 ! This program will copy MEASUREMENT data to a data register. This
50 ! program will also distinguish between IBASIC and BASIC run modes
60 !-----
70 !
90   ON ERROR GOTO Not_ibasic
100  Device=800
110  CLEAR Device
120  GOTO Start_prog
130  !
140 Not_ibasic:
150  Device=711
160 Start_prog:
170  OFF ERROR
180  !
190  !
200  ASSIGN @Analyzer TO Device
210  !
230  OUTPUT @Analyzer;"*RST"
240  OUTPUT @Analyzer;"ABORT;:INIT; *WAI"
250  !
260  OUTPUT @Analyzer;"TRAC D1, TRAC1"
270  !
280  !
290  OUTPUT @Analyzer;"DISP:FORM ULOW"
292  !
294  !
300  OUTPUT @Analyzer;"CALC2:FEED `D1`;MATH:STAT OFF; *WAI"
310  !
330  END

```

Example Programs  
COPYDATA

### Comments for COPYDATA

- 90 - 170    Determines where the program is running; either in the analyzer or on an external controller.
- 100        Specify interface select code and primary address for Instrument BASIC.
- 150        Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 200        Allows device select code to be independent of address.
- 230        Reset the analyzer.
- 240        Start the measurement.
- 260        Copy trace A to data register D1.
- 290        Specify the Upper/Lower display format
- 300        Display D1 in trace B.

## FREQRESP

```

10 ! FREQRESP
20 !-----
30 ! BASIC/IBASIC Program: Freq Response
40 ! This program puts the instrument in a 2-channel mode, sets up a
50 ! frequency response measurement, runs the measurement and then
60 ! transfers the data. This program also distinguishes between IBASIC
70 ! and BASIC run modes.
80 !
90 ! NOTE: for best results connect the source and channel to the input of
100 ! !the device under test, and Channel 2 to the output of the device
110 ! !under test. You could also use a straight connection between
120 ! !both inputs and the source.
130 !-----
140 !
150 ON ERROR GOTO Not_ibasic
160 Device=800
170 CLEAR Device
180 GOTO Start_prog
190 !
200 Not_ibasic:
210 Device=711
220 Start_prog:
230 OFF ERROR
240 !
250 !
260 ASSIGN @Analyzer TO Device
270 !
280 !
290 !
300 DIM A(5000)
310 !
320 OUTPUT @Analyzer;"*RST"
330 OUTPUT @Analyzer;"INP2 on"
340 OUTPUT @Analyzer;"OUTP ON"
350 OUTPUT @Analyzer;"SOUR:FUNC RAND"
360 OUTPUT @Analyzer;"SOUR:VOLT 1 VPK"
370 OUTPUT @Analyzer;"AVER ON"
380 OUTPUT @Analyzer;"ABOR;:INIT; *WAI"
390 OUTPUT @Analyzer;":INIT:CONT OFF"
400 !
410 !
420 ASSIGN @Analyzer_bin TO Device;FORMAT OFF
430 !
440 !
450 ! .
460 OUTPUT @Analyzer;"form:data real,64"
470 !

```

Program listing continues on page F-13.

Example Programs  
FREQRESP

## Comments for FREQRESP

- 150 - 230     Determines where the program is running; either in the analyzer or on an external controller.
- 160     Specify interface select code and primary address for Instrument BASIC.
- 210     Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 260     Allows device select code to be independent of address.
- 300     Reserve a large space for the data.
- 320     Reset the analyzer.
- 330     Specify 2-channel instrument mode.
- 340     Turn on the source.
- 350     Set the source to random noise.
- 360     Set the source level.
- 370     Turn on averaging (default number of averages = 10).
- 380     Start the measurement.
- 390     Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 420     Assign an I/O path for binary transfers.
- 460     Tell the analyzer to output block data in floating- point-binary representation (64-bit binary transfer).

```

480  !
490  OUTPUT @Analyzer;"CALC1:FEED `XFR:POW:RAT 2,1'; *WAI"
500  !
510  !
520  OUTPUT @Analyzer;"TRAC:DATA D1,TRAC1"
530  !
540  !
550  !
560  !
570  !
580  OUTPUT @Analyzer;"TRAC:DATA? D1"
590  !
600  !
610  !
620  !
630  ENTER @Analyzer USING "%,A,D";A$,Digits
640  ENTER @Analyzer USING "%,"&VAL$(Digits)&"D";Num_of_bytes
650  !
660  Num_points=Num_of_bytes DIV 8
670  !
680  !
690  !
700  !
710  REDIM A(Num_points-1)
720  !
730  ENTER @Analyzer_bin;A(*)
740  ENTER @Analyzer;A$
750  !
760  !
770  IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG LOW"
780  !
790  !
800  DIM I$(80)
810  I$=""Data point "", 2D, "" Re: "", MD.DDESZZ, "" Im: "", MD.DDESZZ"
820  FOR I=0 TO 9
830     PRINT USING I$;I;A(I*2);A(I*2+1)
840  NEXT I
850  END

```

Example Programs  
FREQRESP

Comments for FREQRESP (continued)

- 490 Display the frequency response data in trace A.
- 520 Store the frequency response data in data register D1.
- 580 Read the data from the data register D1. Since the data is being read with the TRAC:DATA query, the data will be real and imaginary values from the analyzer—not the values shown on the display.
- 630 Read the header of the data block. Read the “#n” to determine how many digits are in the byte count.
- 640 Read the number of bytes.
- 660 Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.
- 710 Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.
- 730 Read all the numbers in the block into the array in binary.
- 740 Read the line feed character.
- 770 If running the program in Instrument BASIC, open a display window.
- 800 - 840 Print the first ten real and imaginary pairs.



## GETASCII

```

10 ! GETASCII
20 !-----
30 ! BASIC/IBASIC Program: ASCII query
40 ! This program will read back the trace data values (401 data points
50 ! for default FFT analysis) and display the first ten values on the
60 ! screen. This program will also distinguish between BASIC/IBASIC.
70 !-----
80 !
90 ON ERROR GOTO Not_ibasic
100 Device = 800
110 CLEAR Device
120 GOTO Start_prog
130 !
140 Not_ibasic:
150 Device=711
160 Start_prog:
170 OFF ERROR
180 !
190 !
200 ASSIGN @Analyzer TO Device
210 !
220 DIM A(400)
230 OUTPUT @Analyzer;"*RST"
240 OUTPUT @Analyzer;"ABORT;:INIT; *WAI"
250 OUTPUT @Analyzer;":INIT:CONT OFF"
260 !
270 !
280 !
290 OUTPUT @Analyzer;"FORM:DATA ASCII"
300 !
310 !
320 OUTPUT @Analyzer;"CALC1:DATA?"
330 !
340 !
350 IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG LOW"
360 !
370 !
380 FOR I=0 TO 400
390 !
400 !
410 !
420 ENTER @Analyzer USING "%,K";A(I)
430 IF I THEN PRINT "Data point ";I,A(I)
440 NEXT I
450 !
460 END

```

Example Programs  
GETASCII

### Comments for GETASCII

- 90 - 170    Determines where the program is running; either in the analyzer or on an external controller.
- 100    Specify interface select code and primary address for Instrument BASIC.
- 150    Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 200    Allows device select code to be independent of address.
- 220    Dimension the array for the 401 values to be read.
- 230    Reset the analyzer.
- 240    Start the measurement.
- 250    Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 290    Select the ASCII data format. The analyzer will output the block data with number formatted in ASCII character strings.
- 320    Tell the analyzer to output the data from trace A.
- 350    If the program is running in Instrument BASIC, open a display window.
- 380 - 440    Read the data from the analyzer.
- 420    Read in one point of data; “%” in USING image causes BASIC to read the number and not search for <cr><lf>.

## GETBIN

```

10 ! GETBIN
20 !-----
30 ! BASIC/IBASIC Programs; Power spectrum Query in binary
40 ! This program will query the Channel 1 Power Spectrum Data in
50 ! Binary. This program will also distinguish between IBASIC and
55 ! BASIC run modes.
60 !-----
70 !
80   ON ERROR GOTO Not_ibasic
90   Device = 800
100  CLEAR Device
110  GOTO Start_prog
120  !
130 Not_ibasic:
140   Device=711
150 Start_prog:
160   OFF ERROR
170   !
180   !
190   ASSIGN @Analyzer TO Device
200   OUTPUT @Analyzer;"*RST"
210   OUTPUT @Analyzer;"ABOR;:INIT; *WAI"
220   OUTPUT @Analyzer;":INIT:CONT OFF"
230   !
240   !
250   DIM A(5000)
260   !
270   !
280   ASSIGN @Analyzer_bin TO Device;FORMAT OFF
290   !
300   !
302  !
310   OUTPUT @Analyzer;"form:data real,64"
320   !
330   !
340   OUTPUT @Analyzer;"CALC1:DATA?"
350   !
360   !
370   !
380   !
390   ENTER @Analyzer USING "%,A,D";A$,Digits
400   ENTER @Analyzer USING "%," &VAL$(Digits)&"D";Num_of_bytes
410   !
420   Num_points=Num_of_bytes DIV 8
430   !
440   !

```

Program listing continues on page F-19.

Example Programs  
GETBIN

## Comments for GETBIN

- 80 - 160    Determines where the program is running; either in the analyzer or on an external controller.
- 90        Specify interface select code and primary address for Instrument BASIC.
- 140       Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 190       Allows device select code to be independent of address.
- 200       Reset the analyzer.
- 210       Start the measurement.
- 220       Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 250       Reserve a large space for the data.
- 280       Assign an I/O path for binary transfers.
- 310       Tell the analyzer to output block data in floating-point-binary representation (64-bit binary transfer).
- 340       Tell the analyzer to output the data from trace A.
- 390 - 400    Read the header of the data block.
- 390       Read the "#n" to find out how many digits are in the byte count.
- 400       Read the number of bytes.
- 420       Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.

```
450  !
460  ! 470  REDIM A(Num_points-1)
480  !
490  !
500  ENTER @Analyzer_bin;A(*)
510  ENTER @Analyzer;A$
520  !
530  !
540  IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG LOW"
550  !
560  FOR I=0 TO 9
570     PRINT "Data point ";I,A(I)
580  NEXT I
590  END
```

Example Programs  
GETBIN

Comments for GETBIN (continued)

- 470 Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.
- 500 Read the numbers in the block into the array in binary.
- 510 Read the line feed character.
- 540 If the program is running in Instrument BASIC, open a display window.
- 560 - 590 Print the first 10 values.

## GETCAP

```

10 ! GETCAP
20 !-----
30 ! BASIC/IBASIC Program; Time Capture Data Transfer
40 ! This program sets up a time capture measurement and
50 ! reads the time capture data into an array. The time
60 ! capture data is then modified and put back into the
70 ! capture buffer.
80 !
90 ! NOTE: Connect the Agilent 35670A Source output to the
100 ! Channel 1 input before running this program.
110 !-----
120 !
130 DIM Y(1:100,1:1024)
140 !
150 IF POS(SYSTEM$("SYSTEM ID"),"Agilent 35670") THEN
160     Address=800
170 ELSE
180     Address=711
190 END IF
200 !
210 ASSIGN @Analyzer TO Address
220 ASSIGN @Analyzer_bin TO Address;FORMAT OFF
230 !
240 !
250 OUTPUT @Analyzer;"SYST:PRES"
260 OUTPUT @Analyzer;"VOLT1:RANG 2 VPK"
270 OUTPUT @Analyzer;"OUTP ON"
280 OUTPUT @Analyzer;"SOUR:VOLT 2 VPK"
290 OUTPUT @Analyzer;"SOUR:FREQ 100 HZ"
300 OUTPUT @Analyzer;"CALC:FEED `TCAP 1`"
310 OUTPUT @Analyzer;"DISP:WIND1:TRAC:Y:TOP 5; PDIV 1"
320 OUTPUT @Analyzer;"TCAP"
330 !
340 OUTPUT @Analyzer;"*OPC?"
350 ENTER @Analyzer;Opc
360 !
370 !
380 OUTPUT @Analyzer;"FORM REAL,64"
390 !
400 OUTPUT @Analyzer;"SENSE:DATA? TCAP1"
410 ENTER @Analyzer USING "%,A,D";Resp$,I
420 ENTER @Analyzer USING "%,"&VAL$(I)&"D";Bytes
430 !
440 Points=Bytes/8
450 Records=Points/1024.
460 !

```

Program listing continues on page F-23

Example Programs  
GETCAP

## Comments for GETCAP

- 130     Declare an array equal to or larger than the largest time capture we expect to read. It is convenient to use 1024 for one dimension of the array because the analyzer always captures data in 1024 point blocks.
- 150 - 190     Determine if this program is running in the analyzer or an external controller and set the analyzer address accordingly.
- 210     Assign path for formatted communication with the analyzer.
- 220     Assign path for binary communication with the analyzer.
- 250 - 350     Setup the analyzer to output a 100 HZ, 2 Vpk sine wave and capture 10 records of time data.
- 380     Tell the analyzer to send and receive 64 bit/point (8 byte/point) data blocks.
- 400     Send the query asking the analyzer to send the channel 1 time capture data.
- 410-420     Enter the definite block header (a pound sign, the number of characters in the byte count, and the byte count).
- 440-450     Convert the byte count to number of points and then calculate the number of records. There are 8 bytes/point and 1024 bytes/record.
- 470     Redimension the Y data array so that it is the same size as the data we will receive.
- 500     Enter the Y data using the unformatted path.
- 530     Enter the block termination character.
- 560     Square the Y data using the matrix dot operator.
- 590-600     Calculate the maximum absolute value of the Y data and set the time-capture channel 1 range value to the maximum. This will maximize the dynamic range.
- 620     Calculate the number of bytes we will send to the analyzer.
- 630-640     Convert the byte value to a string. We have to use the OUTPUT statement because VAL\$ would create a string with exponential notation for large values of Bytes.
- 650     Calculate the length of the Bytes\$ string.
- 670     Output the mnemonic to send time-capture data and the definite-length block header (# symbol, number of characters in the byte count, and the byte count).
- 690     Output the data block to the analyzer using the unformatted path.
- 700     Output an END to the analyzer to end the data block.



```

470 REDIM Y(1:Records,1:1024)
480 !
490 DISP "Getting data"
500 ENTER @Analyzer_bin;Y(*)
510 DISP
520 !
530 ENTER @Analyzer;Term$
540 !
550 !
560 MAT Y=Y.Y
570 !
580 !
590 Max_val=MAX(MAX(Y(*),ABS(MIN(Y(*))))))
600 OUTPUT @Analyzer;"SENSE:DATA:RANGE TCAP1, "&VAL$(Max_val)
610 !
620 Bytes=Records*1024.*8.
630 OUTPUT Bytes$ USING "#,9D";Bytes
640 Bytes$=TRIM$(Bytes$)
650 Length=LEN(Bytes$)
660 !
670 OUTPUT @Analyzer;"SENSE:DATA TCAP1,#"&VAL$(Length)&Bytes$;
680 DISP "Sending data"
690 OUTPUT @Analyzer_bin;Y(*)
700 OUTPUT @Analyzer;CHR$(10) END
710 !
720 !
730 DISP "Done"
740 !
750 END

```

## GETCURF

```
10 ! GETCURF
20 !-----
30 ! BASIC/IBASIC Programs; Get Curve Fit table
40 ! This program will query the Curve Fit Data in Binary and print out
50 ! most of the information on the CRT screen. The program does not cover
52 ! all cases but uses the default conditions. The program assumes that
54 ! the Curve Fit has already been run, and the user wishes to dump the
56 ! data out of the Instrument and into the controller. The program also
58 ! prints out most of the acquired data. Use the descriptions found in
60 ! the CALC:CFIT:DATA to extract other available information from the
70 ! Curve Fit table.
80 ! This program will also distinguish between IBASIC and BASIC.
90 !-----
92 !
94     ON ERROR GOTO Not_ibasic
100 Device = 800
110 CLEAR Device
120 GOTO Start_prog
130 !
140 Not_ibasic:
150     Device=711
160 Start_prog:
170     OFF ERROR
250     !
260     ASSIGN @Analyzer TO Device     !
270     DIM A(5000)
280     ASSIGN @Analyzer_bin TO Device;FORMAT OFF
290     OUTPUT @Analyzer;"FORM:DATA REAL,64"
300     !
320     OUTPUT @Analyzer;"CALC1:CFIT:DATA?"
340     !
342     !
344     !
346     !
350     ENTER @Analyzer USING "%,A,D";A$,Digits
390     ENTER @Analyzer USING "%,"&VAL$(Digits)&"D";Num_of_bytes
400     Num_points=Num_of_bytes DIV 8
410     !
430     REDIM A(Num_points-1)
440     ENTER @Analyzer_bin;A(*)
450     ENTER @Analyzer;A$
```

Program listing continues on page F-26.

## Comments for GETCURF

- 94 - 170    Determines where the program is running; either in the analyzer or on an external controller.
- 100    Specify interface select code and primary address for Instrument BASIC.
- 150    Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 260    Allows device select code to be independent of address.
- 270    Reserve a large space to read data.
- 280    Assign an I/O path for binary transfers.
- 290    Tell the analyzer to output block data in floating-point-binary representation (64-bit binary transfer).
- 320    Tell the analyzer to output the curve fit data. Use CALC1:SYNT:DATA? to extract the synthesis table data.
- 350    Read the header of the data block. Read the "#n" to determine how many digits are in the byte count.
- 390    Read the number of bytes.
- 400    Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.
- 430    Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.
- 440    Read the data in the curve fit table in binary.
- 450    Read the line feed character.

## Example Programs

### GETCURF

```
460  !
470  !
480  IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG FULL"
490  !
500  !
510  PRINT ""
520  IF A(0)=0 THEN PRINT "TABLE TYPE = POLE ZERO"
530  IF A(0)=1 THEN PRINT "TABLE TYPE = POLE RESIDUE"
540  IF A(0)=2 THEN PRINT "TABLE TYPE = POLYNOMIAL"
550  PRINT ""
560  PRINT "          POLES"
570  PRINT "          REAL          IMAGINARY"
580  FOR I=4 TO 4+2*(A(1)-1) STEP 2
590    PRINT USING "10D.4D,3X,5A,7D.3D";A(I),"+/- j",A(I+1)
600  NEXT I
610  PRINT ""
620  PRINT "          ZEROS"
630  PRINT "          REAL          IMAGINARY"
640  FOR I=46 TO 46+2*(A(2)-1) STEP 2
650    PRINT USING "10D.4D,3X,4A,7D.3D";A(I),"+/- j",A(I+1)
660  NEXT I
670  PRINT ""
680  PRINT ""
690  PRINT "CURVE FIT TERMS"
700  PRINT "          POLES          ZERO'S"
710  PRINT ""
720  FOR I=130 TO 130+(A(1)-1) STEP 1
730    IF A(I)=0 THEN
740      PRINT USING "8X,10A,#";"MOVEABLE"
750    ELSE
760      PRINT USING "8X,10A,#";"FIXED"
770    END IF
780    IF A(I+21)=0 THEN
790      PRINT USING "8X,10A";"MOVEABLE"
800    ELSE
810      PRINT USING "8X,10A";"FIXED"
820    END IF
830  NEXT I
840  PRINT ""
850  PRINT " GAIN = ",A(Num_points-3)
860  PRINT " FREQUENCY SCALE = ",A(Num_points-2),"HZ/HZ"
870  PRINT " TIME DELAY = ",A(Num_points-1),"SECONDS"
880  END
```

### Comments for GETCURF (continued)

- 480    If running the program in Instrument BASIC, open a display window.
- 510 - 570    Print the header information of the curve fit table.
- 580 - 630    Print the left side of the table.
- 640 - 710    Print the right side of the table.
- 720 - 830    Print whether terms are “fixed” or moveable.
- 840 - 870    Print the final parameters of the curve fit table.

## GETWATR

```
10 ! GETWATR
20 !-----
30 ! BASIC/IBASIC Programs; Octave Waterfall Dump
40 !
50 ! this program is divided into three parts. (1) set up the source, (2) set up
60 ! and take a MEASUREMENT, (3) dump the data out of the instrument and print
70 ! out the information. The information is in dBSLPrms units
80 !
90 ! Some items of note.
100 !! 1. default is 1 engineering unit per volt
110 !! 2. engineering unit is defined in Pascals
120 !! 3. 20E-6 pascals = 1 SPL or 0 dBVrms = approx 94 dBSPLrms
130 !! 4. waterfall can only be dumped if the MEASUREMENT is PAUSED
140 !! 5. Connect the SOURCE to CHANNEL 1
150 !! 6. Overall power and A-weight Filter data is always outputted
160 !! 7. Data in the array A(*) is from Oldest to newest data.
170 !! 8. This is the opposite of the display which has oldest on
180 !! !the bottom
190 !
200 ! This program will also distinguish between IBASIC and BASIC run
201 ! Modes.
210 !-----
220 ON ERROR GOTO Not_ibasic
230 Device = 800
240 CLEAR Device
245 CLEAR SCREEN
250 GOTO Start_prog
260 !
270 Not_ibasic:
280 Device=711
290 Start_prog:
300 OFF ERROR
310 !
320 ASSIGN @Analyzer TO Device
330 !
340 !
350 DIM A(20000),X(100)
360 !
370 !
380 OUTPUT @Analyzer;"*RST"
390 OUTPUT @Analyzer;"INST OCT"
395 OUTPUT @Analyzer;"FREQ:RES:OCT THIR"
400 ! OUTPUT @Analyzer;"FREQ:RES:OCT FULL"
410 ! OUTPUT @Analyzer;"FREQ:RES:OCT TWEL"
420 !
```

Program listing continues on page F-30.

## Comments for GETWATR

- 220 - 300     Determines where the program is running; either in the analyzer or on an external controller.
- 230     Specify interface select code and primary address for Instrument BASIC.
- 280     Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 320     Allows device select code to be independent of address.
- 350     Reserve a large space for the frequency and amplitude data.
- 380     Reset the analyzer.
- 390     Set instrument mode to realtime octave analysis.
- 395     Specify 1/3 octave band measurement.
- 400     Specify 1/1 octave band measurement. Uncomment the statement (delete the "!") to specify this type of measurement.
- 410     Specify 1/12 octave band measurement. Uncomment the statement (delete the "!") to specify this type of measurement.

## Example Programs

### GETWATR

```
430 OUTPUT @Analyzer;"OUTP ON"
440 OUTPUT @Analyzer;"SOUR:VOLT 1 VRMS"
450 OUTPUT @Analyzer;"SOUR:FREQ 1000 HZ"
460 !
470 OUTPUT @Analyzer;"DISP:FORM ULOW"
480 OUTPUT @Analyzer;"DISP:WIND1:WAT ON"
490 !
496 OUTPUT @Analyzer;"DISP:WIND1:TRAC:APOW ON"
500 !
510 !
520 OUTPUT @Analyzer;"CALC1:FEED `XFR:POW 1`;MATH:STAT OFF;*WAI"
530 !
540 !
550 OUTPUT @Analyzer;"VOLT1:RANG:UNIT:XDCR:LAB PA"
560 OUTPUT @Analyzer;"VOLT1:RANG:UNIT:USER ON"
570 !
580 !
590 OUTPUT @Analyzer;"ABOR;:INIT;*WAI"
600 !
610 !
620 OUTPUT @Analyzer;"INIT:CONT OFF; *WAI"
630 !
640 !
650 !
660 !
670 !
680 ASSIGN @Analyzer_bin TO Device;FORMAT OFF
690 !
700 !
710
720 OUTPUT @Analyzer;"FORM:DATA REAL,64"
740 !
750 !
760 OUTPUT @Analyzer;"CALC1:X:DATA?"
770 !
780 !
790 !
800 !
810 ENTER @Analyzer USING "%,A,D";A$,X_digits
820 !
830 ENTER @Analyzer USING "%,"&VAL$(X_digits)&"D";Num_of_bytes
840 X_num_points=Num_of_bytes DIV 8
850 !
860 !
870 !
880
890 REDIM X(X_num_points-1)
```

Program listing continues on page F-32.



## Comments for GETWATR (continued)

- 430 Turn on the source. The default type is a fixed sinusoidal waveform.
- 440 Set the source level.
- 440 Set the frequency of the sine source to 1000 Hz.
- 480 Turn on the waterfall display for trace A.
- 496 Display the weighted power band.
- 520 Specify Channel 1 power spectrum data in trace A waterfall display.
- 550 Label the transducer unit PA (Pascals) for Channel 1.
- 560 Enable the use of the transducer unit.
- 590 Start a measurement. Wait for the measurement to complete collecting data.
- 620 Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 640 - 1100 Transfer the frequency and amplitude data to an array. Frequency data (X-axis values) and amplitude data (Y-axis values) are transferred separately. Frequency data is transferred first.
- 680 Assign an I/O path for binary transfers.
- 720 Tell the analyzer to output block data in floating-point-binary representation (64-bit binary transfer).
- 760 Tell the analyzer to output the X-axis values for trace A.
- 810 - 830 Read the header of the X-axis values data block.
- 810 Read the “#n” to determine how many digits are in the byte count.
- 830 Read the number of bytes.
- 840 Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.
- 890 Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.

## Example Programs

### GETWATR

```
900  !
910  !
920  ENTER @Analyzer_bin;X(*)
930  ENTER @Analyzer;A$
940  !
950  !
960  !
970  OUTPUT @Analyzer;"CALC1:WAT:DATA?"
980  !
990  !
1000 ENTER @Analyzer USING "%,A,D";A$,Y_digits
1010 ENTER @Analyzer USING "%,"&VAL$(Y_digits)&"D";Num_of_bytes
1020  !
1030 Y_num_points=Num_of_bytes DIV 8
1040  !
1050  !
1060 REDIM A(Y_num_points-1)
1070  !
1080  !
1090 ENTER @Analyzer_bin;A(*)
1100 ENTER @Analyzer;A$
1110  !
1120  !
1130  !
1140  !
1150  !
1160 FOR I=0 TO Y_num_points-1
1170     IF A(I)99 THEN A(I)=-99
1172     IF A(I)99 THEN A(I)=99
1180 NEXT I
1190  !
1200  !
1210 IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG LOW"
1220  !
1230  !
1240  !
1250  !
```

Program listing continues on page F-34.

### Comments for GETWATR (continued)

- 920 Read the X-axis values in the block into the array in binary.
- 930 Read the line feed character.
- 970 Tell the analyzer to output the Y-axis values for the waterfall.
- 1000 - 1010 Read the header of the Y-axis values data block.
- 1000 Read the “#n” to determine how many digits are in the byte count.
- 1010 Read the number of bytes.
- 1030 Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.
- 1060 Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.
- 1090 Read all the Y-axis values in the block into the array in binary.
- 1100 Read the line feed character.
- 1160 - 1180 Limits the range of values. When an octave band has no energy, the value is - ∞ dB and the large negative number (-3.40E38) appears in the data block. These statements restrict the image statements to the ± 1000 range.
- 1160 Start the array at 0.
- 1170 Any values < -99, set to -99. This sets a minimum level.
- 1180 Any values > -99, set to -99. This sets a minimum level.
- 1210 If running the program in Instrument BASIC, open a display window.

## Example Programs

### GETWATR

```
1260 PRINT "CENTER FREQ          SCAN1          SCAN2          SCAN3"
1270 PRINT
1280 FOR I=0 TO X_num_points-3
1290 Im_1:  IMAGE  7D.D,11X,M4D.DD,3X,M4D.DD,3X,M4D.DD
1300 PRINT USING Im_1;X(I),A(I),A(I+X_num_points),A(I+2*X_num_points)
1301 IF Device=800 AND (I MOD 10=9) THEN
1302     BEEP 200,.2
1303     DISP "press continue to see more values"
1304     PAUSE
1305     DISP ""
1306 END IF
1310 NEXT I
1320 PRINT
1330 Im_2:  IMAGE  16A,4X,M4D.2D,3X,M4D.2D,3X,M4D.2D
1340 PRINT USING Im_2;" A WEIGHT DATA          "
,A(X_num_points-2),A(2*X_num_points-2),A(3*X_num_points-2)
1350 PRINT USING 1330;" OVERALL POWER          "
,A(X_num_points-1),A(2*X_num_points-1),A(3*X_num_points-1)
1360 END
```

Comments for GETWATR (continued)

1260 - 1350    The print routines.

1260    Print the column headings

1280 -1310    Print the amplitude values in the first three “scans” for each frequency. A “scan” is a trace in the waterfall.

1304    Pause the program for each group of 10 frequencies.

1330 - 1350    Print the A-weighted and overall band values for the first three scans.

## TIMEBIN

```
10 ! TIMEBIN
20 !-----
30 ! BASIC/IBASIC Programs; Time Query
40 ! This program will query the Channel 1 Time Data in Binary. This
50 ! program will also distinguish between IBASIC and BASIC run modes.
60 !-----
70 !
80   ON ERROR GOTO Not_ibasic
90   Device = 800
100  CLEAR Device
110  GOTO Start_prog
120  !
130 Not_ibasic:
140   Device=711
150 Start_prog:
160   OFF ERROR
170   !
180   !
190   ASSIGN @Analyzer TO Device
200   OUTPUT @Analyzer;"*RST"
210   OUTPUT @Analyzer;"ABOR;:INIT; *WAI"
220   OUTPUT @Analyzer;":INIT:CONT OFF"
230   !
240   !
250   DIM A(5000)
260   !
270   !
280   ASSIGN @Analyzer_bin TO Device;FORMAT OFF
290   !
300   !
310
320   OUTPUT @Analyzer;"form:data real,64"
330   !
340   ! display Channel 1 time data in trace A
350   OUTPUT @Analyzer;"CALC1:FEED `XTIM:VOLT 1`;MATH:STAT OFF; *WAI"
360   !
370   !
380   OUTPUT @Analyzer;"CALC1:DATA?"
390   !
400   !
410   !
420   !
430   ENTER @Analyzer USING "%,A,D";A$,Digits
440   ENTER @Analyzer USING "%,"&VAL$(Digits)&"D";Num_of_bytes
450   !
460   Num_points=Num_of_bytes DIV 8
470   !
```

Program listing continues on page F-38.

## Comments for TIMEBIN

- 80 - 160    Determines where the program is running; either in the analyzer or on an external controller.
- 90        Specify interface select code and primary address for Instrument BASIC.
- 140       Specify analyzer's interface select code and primary address when the program is running on an external computer. The specified primary address, 11, is the address set at the factory. If the address for your analyzer has changed, edit the primary address to the appropriate value.
- 190       Allows device select code to be independent of address.
- 20        Reset the analyzer.
- 210       Start the measurement.
- 220       Pause the analyzer after the first measurement. This helps speed-up the execution of the program. It also helps speed-up the transfer of data between the program and the analyzer.
- 250       Reserve a large space for the data.
- 280       Assign an I/O path for binary transfers.
- 320       Tell the analyzer to output block data in floating-point-binary representation (64 bit binary transfer).
- 350       Display Channel 1 time data in trace A.
- 380       Tell the analyzer to output the data from trace A.
- 430 - 440    Read the header of the data block.
- 430       Read the "#n" to find out how many digits are in the byte count.
- 440       Read the number of bytes.
- 460       Use the number of bytes to determine the number of data points. Divide the number of bytes by 8.

## Example Programs

### TIMEBIN

```
480  !
490  !
500  !
510  REDIM A(Num_points-1)
520  !
530  !
540  ENTER @Analyzer_bin;A(*)
550  ENTER @Analyzer;A$
560  !
570  !
580  IF Device=800 THEN OUTPUT @Analyzer;"DISP:PROG LOW"
590  !
600  FOR I=0 TO 9
610      PRINT "Data point ";I,A(I)
620  NEXT I
630  END
```



### Comments for TIMEBIN (continued)

- 510 Re-dimension the array so it is the same size as the data block being read. This allows the data to be read in the fastest possible manner.
- 540 Read the numbers in the block into the array in binary.
- 550 Read the line feed character.
- 580 If the program is running in Instrument BASIC, open a display window.
- 600 - 620 Print the first ten values.

## XFER

File: xfer.c

This C program demonstrates how to do block transfers between a PC and the Agilent 35670A analyzer. This program can be used to transfer time-capture files from the 35670A to a PC file.

The compiled version of this program is on the Example Program disc in the file named XFER.EXE.

The command line syntax for XFER.EXE is:

XFER <file>	[/U] [/I] [/A] [/S] [/C] [/D][ /O]
<File>	path and file name
/U	show this usage
/I: <code>	Select code (default = 7 for Agilent Technologies card<E>, 0 for National Instruments card)
/A: <add>	Analyzer address (default = 11)
/S	Send <file> to analyzer
/C:<cmd>	Analyzer command string (default = TCAP:FILE)
/D:<H N>	H = use Agilent Technologies GPIB card (default) N = use National Instruments GPIB card
/O	Overwrite <file> if it exists.

For example, to get the time capture buffer from the Agilent 35670A to a filenameed CAPT1.TIM on a PC using a Agilent Technologies card, use this command:

```
xfer CAPT1.TIM
```

To send the CAPT1.TIM file back to the Agilent 35670A using a National AT-GPIB card, execute this command:

```
xfer /S /D:N CAPT1.TIM
```

This program supports the Agilent 82335A card and the National Instruments AT-GPIB card. To compile it you need the clGPIB.lib library for the Agilent card and the mcib.obj library for the NI card.

The source code for the XFER program is on the example programs disc in the XFER.C file. The executable file is on the example programs disc in the XFER.EXE file.

# INDEX

---



# INDEX

---

## A

### aborting

- a curve fit operation 6-4
- a measurement 4-2
- a time capture 18-68

### AC power 21-22

### active controller 3-8

### active program 17-10

### active trace 6-3

### address

- Agilent 35670A 21-4
- Agilent\~35670A 3-3
- plotter 10-23
- printer 10-24

### addressable-only 3-4, 3-8

### amplitude

- unit for Y-axis 6-111
- vertical unit 6-119

### analyzer

- fan 21-14
- identification 3-6
- options 3-9
- power setting 21-22
- turning off (via GPIB) 21-23

### anti-alias filter 12-5

### arbitrary source

- low-pass filter 16-2
- selecting 19-7

### arm

- automatic 5-6
- manual 5-2, 5-6
- RPM 5-6
- setting steps 5-3
- Start RPM Arming qualifier 5-4
- starting RPM value 5-5
- step size 5-8
- SEE ALSO tachometer
- time step 5-6, 5-8
- SEE ALSO trigger
- types 5-6
- Waiting for ARM bit 1-23

### ASCII, data transfer format 9-2

### \*CAL 3-2

### \*CLS 3-3

### \*ESE 3-4

### \*ESR? 3-5

### \*IDN? 3-6

### \*OPC 3-7

### \*OPC? 3-7

### \*OPT? 3-9

### \*PCB 3-10

### \*PSC 3-11

### \*RST 3-12

### \*SRE 3-14

### \*STB? 3-16

### \*TRG 3-18

### \*TST? 3-19

### \*WAI 3-20

### Autocal Off bit 1-16

### autocalibration 7-3

### autocorrelation 6-24

### autolevel

- enabling 19-9
- non-reference amplitude 19-16
- selecting reference channel 19-13
- setting amplitude 19-12
- setting limit 19-15
- tolerance 19-14

### autoranging 18-79, 18-81

### autoscaling

- horizontal 8-30
- vertical 8-35

### averaging

- autostart 18-14
- Averaging bit 1-23
- count 18-3
- enabling 18-12
- equal confidence 18-2, 18-19
- exponential 18-14, 18-19
- linear 18-14, 18-19
- normal 18-14
- overload rejection 18-59
- peak hold 18-19
- power 18-19
- preview 18-8 - 18-11
- repeat 18-14
- RMS 18-19
- termination control 18-14
- time 18-17
- transition of bits 18-13
- type of hold in octave 18-4
- valid for instrument mode 18-15
- vector 18-19
- weighting 18-3

### A-weight filter

- applied to input 12-4
- as a math function 6-97

## INDEX (Continued)

- determining 6-70
- B**
- bandwidth 18-40
- BASIC
  - SEE Instrument BASIC
- beeper
  - limit test 6-33
  - main 21-2 - 21-3
- blanking
  - annotation 8-2
  - display 8-15
  - grid 8-26
- blanking:RPM indicator 8-13
- block data
  - number of points 6-20, 6-22
  - X-axis values 6-138
- Bode diagram 8-3
- buffer, defined 3-10
- bus trigger 24-5
- B-weight filter 6-97
- C**
- \*CAL 3-2
- calculate data
  - format 6-29
  - SEE ALSO trace data
  - waterfall 6-128
  - SEE ALSO waterfall display
  - Y-axis scaling 8-42
- Calibrating bit 1-23
- calibration
  - automatic 7-3
  - single 7-3
  - test 3-2, 7-2
- Calibration bit 1-18
- calibration data
  - displaying register 6-23
  - storing to register 23-2
- capture
  - SEE time capture
- catalog 8-17
- CDF 6-25
- center frequency 18-29
- channels
  - reference 18-58
  - specifying 12-9
- clearing
  - screen output 8-11
  - test log 22-2
- clock
  - setting date 21-12
  - setting time 21-26
- \*CLS 3-3
- command reference
  - conventions 2-6
  - description 2-1
  - finding a command 2-3 - 2-4
  - symbols 2-5
  - syntax descriptions 2-7
- command summary A-1 - A-20
- composite power
  - calculation 18-89
  - displaying 6-26
- condition register
  - Device State 20-2
  - Limit Fail 20-16
  - Operation Status 20-7
  - Questionable Status 20-13
  - Questionable Voltage 20-23
  - transition of bits 18-13
- configuring the GPIB system 3-2
- constant
  - defining 6-94
  - displaying 6-99
- continuing a program 17-12
- continuous trigger 24-8
- controller
  - SEE ALSO active controller
  - configuring 3-2
  - SEE ALSO system controller
- controller capabilities 3-4, 3-8
- coordinates, selecting 6-29
- copy
  - data tables 6-58
  - disk 15-3
  - file 15-3
- correlation analysis
  - instrument mode 13-2 - 13-3
  - weighting function 18-91
- correlation weighting function
  - 0, T/2 18-91
  - T/4, T/4 18-91
  - uniform 18-91
- coupling
  - input 12-3
  - markers 6-56
- cross correlation 6-24
- cross spectrum 6-24
- cumulative density function, displaying 6-25
- curve fit
  - number of poles 6-15
  - copy 6-5
  - SEE ALSO curve fit table
  - executing 6-13
  - fit region 6-9 - 6-11
  - frequency scaling 6-12
  - limiting region 6-9
  - loading values via GPIB 6-6

- number of zeros 6-16
- operation 6-14
- selecting data register 6-8
- setting orders 6-14
- setting region 6-10 - 6-11
- starting 6-13
- storing to a file 15-26
- time delay 6-17
- weighting 6-18 - 6-19
- curve fit table
  - displaying 8-17
  - loading from a file 15-10
  - storing to a file 15-26
  - transferring via GPIB 6-6
- C-weight filter 6-97
- D**
- data points
  - specifying time record 18-27
- data register
  - arbitrary source data 19-7
  - copying a waterfall trace 6-135
  - curve fit results 6-8
  - displaying 6-23
  - selecting for synthesis 6-104
  - storing a waterfall slice 6-131
  - storing trace data 23-2
  - SEE ALSO waterfall register
  - weighting function for curve fit 6-19
- data tables
  - clearing 6-57
  - copying 6-58
  - deleting entries 6-62
  - displaying 8-19
  - inserting entries 6-63
  - labeling 6-64
  - loading from disk 15-12
  - markers 8-18
  - selecting entries 6-65
  - storing from disk 15-28
  - transferring 6-59, 6-61
- data transfer format
  - ASCII 9-2
  - real 9-2
- date 21-12
- dB reference 6-114 - 6-117
- DC offset 19-11
- DC power 21-22
- default disk 15-24
- degree units 6-113
- delete
  - data table entries 6-62
  - disk 15-4
  - file 15-4
  - lower limit lines 6-35
  - memory 14-4 - 14-5
  - program 17-6 - 17-7
  - time capture buffer 18-69
  - upper limit lines 6-44
- device reset 3-12
- Device State register set
  - condition register 20-2
  - defined 1-16
  - enable register 20-3
  - event register 20-4
  - negative transition register 20-5
  - positive transition register 20-6
- directory
  - creating 15-22
- disabling the keyboard 21-21
- disk
  - addressing 15-5 - 15-6
  - catalog 8-17
  - copying 15-3
  - creating a directory 15-22
  - data tables 15-12, 15-28
  - default 15-24
  - deleting 15-4
  - file system 15-7
  - formatting 15-8
  - initializing 15-8
  - loading math definitions 15-15
  - loading multiple 15-11
  - lower limit lines 15-13, 15-29
  - storing math definitions 15-31
  - storing multiple 15-27
  - upper limit lines 15-14, 15-30
- disk operations, format 15-9
- display
  - adjusting brightness 8-4
  - Bode diagram 8-3
  - displaying graticules 8-26
  - external monitor 8-6
  - fast average mode 18-7
  - first X-axis value 8-31
  - formatting text 8-5
  - group delay 6-30
  - last X-axis value 8-32
  - number of data points 18-27
  - number of orders 18-51
  - Nyquist diagram 6-30
  - overall band 8-24 - 8-25
  - partitioning for Instrument BASIC 8-11
  - plotting 10-4, 10-26
  - polar diagram 6-30, 8-21 - 8-22
  - printing 10-4
  - reference level 8-36 - 8-37, 8-39, 8-41
  - reference level tracking 8-39
  - RPM indicator 8-13
  - scaling 8-31 - 8-32
  - selecting contents 8-17
  - selecting format 8-7
  - setting for Instrument BASIC 8-9 - 8-10

## INDEX (Continued)

- time capture envelope 8-16
- trace label 8-27 - 8-28
- turning on 8-15
- updates 18-7
- waterfalls 8-43 - 8-51
- Y-axis amplitude unit 6-111
- Y-axis units E-1 - E-6
- Y-axis vertical unit 6-119
- display data
  - SEE calculate data
  - SEE ALSO trace data
- downloading data
  - SEE ALSO example programs
  - SEE loading
- downloading files
  - SEE loading
- E**
- echoing mnemonics 8-8
- enable register
  - Device State 20-3
  - effect by preset 20-12
  - Limit Fail 20-17
  - Operation Status 20-8
  - Questionable Status 20-14
  - Questionable Voltage 20-24
  - User Status 20-28
- ^ END 2-6
- End or Identify (EOI) 2-6
- energy spectral density units 6-119
- engineering units
  - enabling 18-84
  - labeling 18-82
  - scaling 18-83
- EOI 2-6
- error messages
  - displaying 8-5, 8-17
  - listing C-1 - C-11
  - reading 21-13
- ESD units 6-119
- \*ESE 3-4
- \*ESR? 3-5
- event register
  - Device State 20-4
  - Limit Fail 20-18
  - Operation Status 20-9
  - Questionable Status 20-15
  - Questionable Voltage 20-25
  - User Status 20-29 - 20-30
- example programs
  - arbitrary source F-3 - F-8
  - copying data F-9 - F-10
  - downloading data F-3 - F-8
  - frequency response F-11 - F-14
  - reading curve fit data F-24 - F-27
  - reading time data F-36 - F-39
  - reading trace data F-15 - F-20
  - reading waterfall data F-28 - F-35
  - transferring data (ASCII) F-15 - F-16
  - transferring data (binary) F-11 - F-14, F-17 - F-20, F-24 - F-39
  - transferring data (curve fit) F-24 - F-27
  - transferring data (octave) F-28 - F-35
  - transferring data (waterfall) F-28 - F-35
  - using data registers F-3 - F-10
- exponential window 18-90
- external controller
  - SEE ALSO example programs
  - loading program 17-3, 17-5
  - loading program label 17-4
- external disk
  - address 15-5
  - unit 15-6
- external monitor 8-6
- external trigger
  - filter 24-2
  - level 24-3
  - range 24-4
  - selecting 24-8
- F**
- fan 21-14
- fast average mode
  - enabling 18-7
  - rate 18-6
- fault log
  - clearing 21-15
  - displaying 8-17
- FFT analysis instrument mode 13-2 - 13-3
- files
  - copying 15-3
  - deleting 15-4
  - SEE ALSO loading
  - naming plot output 15-25
  - naming print output 15-25
  - renaming 15-23
  - SEE ALSO storing
- filter
  - A-weight 6-97, 12-4
  - B-weight 6-97
  - C-weight 6-97
  - for external trigger 24-2
- flat top window 18-90
- floating an input 12-6
- force window 18-90
- format
  - data transfer 9-2
  - options for disk 15-9
  - trace display 8-7
- formatting disks 15-8
- frequency



- bandwidth 18-40
- center 18-29
- full span 18-43
- link 18-45
- manual sweep 18-30
- resolution 18-31, 18-40
- sine source type 19-4
- span 18-40
- start 18-46
- step size 18-48
- step size (setting) 18-31
- stop 18-49
- frequency response function 6-24
- frequency scaling
  - curve fit 6-12
  - synthesis 6-105
- FRF 6-24
- front panel
  - keycodes 21-17 - 21-19
  - keys versus GPIB commands B-1
  - keystroke echo 1-38
  - preset 21-24
- full span 18-43
- function
  - defining 6-97
  - displaying 6-99
- fundamental frequency, specifying 6-74
- G**
- gain constant 6-106
- go-no go testing
  - SEE limit test
- GPIB
  - SEE GPIB
- GPIB echo 8-8
- graticule
  - assigning plotter pen 10-15
  - displaying 8-26
  - plotting 10-26
- grounding an input 12-6
- group delay
  - aperture 6-31
  - displaying 6-30
- H**
- Hanning window 18-90
- harmonic markers
  - determining power 6-66
  - determining total harmonic distortion (THD) 6-66
  - fundamental frequency 6-74
  - SEE ALSO marker functions
  - number of harmonics 6-73
- histogram
  - averaging 18-17
  - cumulative density function 6-25
  - displaying 6-25
  - instrument mode 13-2
  - number of bins 18-50
  - probability density function 6-25
  - unfiltered time record 6-25
- histogram instrument mode 13-3
- Agilent 35670A
  - compliance to SCPI 1-26 - 1-38
  - fan 21-14
  - identification 3-6
  - instrument modes 13-2 - 13-3
  - option configuration 3-9
  - register set summary 1-25
  - register sets 1-13 - 1-25
- Instrument BASIC
  - active program 17-10
  - allocating memory space 17-9
  - analyzer's power 21-23
  - buffering of lines 8-12
  - continuing a program 17-12
  - deleting program 17-6 - 17-7
  - editor 17-2
  - labeling program 17-8
  - labeling via GPIB 17-4
  - loading label from external controller 17-4
  - loading program from external controller 17-3, 17-5
  - loading programs from disk 15-16
  - partitioning display for 8-11
  - pausing a program 17-12
  - Program Running bit 1-23
  - purging 14-5
  - reading and writing numeric variables 17-11
  - reading and writing string variables 17-13
  - running a program 17-12
  - saving a program 15-32 - 15-33
  - selecting a program 17-10
  - setting display for 8-9 - 8-10
  - stopping a program 17-12
  - storing programs (format) 15-33
  - storing programs to disk 15-32
  - transferring via GPIB 17-3, 17-5
- GPIB
  - address 21-4
  - command summary A-1 - A-20
  - commands versus front-panel keys B-1
  - configuring the system 3-2
  - echoing commands 8-8
  - error messages C-1 - C-11
  - interface capabilities 3-9
  - setup 3-2 - 3-7
  - trigger 24-8
- I**
- identifying the analyzer 3-6
- \*IDN 3-6
- imaginary component 6-29

## INDEX (Continued)

- impulse detection 18-5
- input
  - AC coupling 12-3
  - autoranging 18-79, 18-81
  - DC coupling 12-3
  - enabling the anti-alias filter 12-5
  - enabling the A-weight filter 12-4
  - enabling the ICP supply 12-2
  - filters 6-70
  - float 12-6
  - ground 12-6
  - reference channel 18-58
  - setting range manually 18-86
  - source for measurement data 18-26
  - specifying channels 12-9
  - trigger level 24-6
- inserting entries, data tables 6-63
- Instrument BASIC
  - SEE Instrument BASIC
- instrument mode
  - selecting 13-2 - 13-3
  - valid GPIB commands D-1 - D-11
  - valid types of averaging 18-15
- instrument state
  - displaying 8-17
  - loading from disk 15-17
  - storing to disk 15-34
  - transferring via GPIB 21-25
- instrument-specific information
  - address 3-3
  - buffers 3-10
  - configuring the system 3-2
  - controller capabilities 3-4, 3-8
  - GPIB setup 3-2 - 3-7
  - interface capabilities 3-9
  - message syntax 2-5 - 2-11
  - overlapped commands 3-10
  - queues 3-10
  - quick verification 3-5
  - status groups 1-13 - 1-25
  - synchronization 3-10
  - syntax conventions 2-6
  - verification program 3-7
- integration time 18-61
- interface capabilities 3-9
- K**
- key presses 21-16
- keyboard, disabling 21-21
- L**
- labeling
  - data tables 6-64
  - SEE ALSO naming
  - softkey 17-4, 17-8
- Limit Fail register set
  - condition register 20-16
  - defined 1-17
  - enable register 20-17
  - event register 20-18
  - negative transition register 20-19
  - positive transition register 20-20
- limit lines
  - assigning line type 10-16
  - assigning plotter pens 10-18
  - defining lower segments 6-39
  - defining upper segments 6-48
  - deleting lower lines 6-35
  - deleting upper lines 6-44
  - deleting upper segment 6-50
  - enabling 8-20
  - SEE ALSO limit test
  - loading lower limits from disk 15-13
  - loading upper limits to disk 15-14
  - moving lower lines 6-36
  - moving upper lines 6-45
  - reporting failed points 6-46 - 6-47
  - storing lower limits to disk 15-29
  - storing upper limits to disk 15-30
  - using the active trace 6-42, 6-51
- limit test
  - beeper 6-33
  - displaying limits 8-20
  - enabling 6-43
  - SEE ALSO limit lines
  - reporting failed points 6-37 - 6-38
  - result 6-34
- line feed character (NL) 2-6
- line types
  - defining 10-16 - 10-17
  - SEE ALSO plotter
- linear spectrum function 6-23
- loading
  - curve fit table (from disk) 15-10
  - curve fit table (via GPIB) 6-6
  - data tables (from disk) 15-12
  - SEE ALSO example programs
  - instrument state (from disk) 15-17
  - instrument state (via GPIB) 21-25
  - lower limits (from disk) 15-13
  - lower limits (via GPIB) 6-39
  - math definitions (from disk) 15-15
  - math definitions (via GPIB) 6-95
  - programs (from disk) 15-16
  - programs (via GPIB) 17-3, 17-5
  - synthesis table (from disk) 15-18
  - synthesis table (via GPIB) 6-102
  - time capture (from disk) 15-19
  - time capture files (large) 15-11
  - trace data (from disk) 15-20
  - trace data (register) 23-2
  - trace data (via GPIB) 23-2
  - upper limits (from disk) 15-14

- upper limits (via GPIB) 6-48
- waterfall data (register) 23-5
- waterfall data (via GPIB) 23-5
- waterfall files (from disk) 15-21
- long confidence test
  - executing 22-3
  - result 22-4
- M**
- magnitude 6-29
- manual sweep
  - enabling 18-62
  - selecting frequency 18-30
- marker
  - assigning plotter pens 10-18
  - coupling 6-56
  - data tables 8-18
  - disabling all 6-89
  - enabling main marker 6-89
  - independent axis position 6-80
  - main marker X-axis position 6-90
  - main marker Y-axis position 6-92
  - SEE ALSO marker functions
  - SEE ALSO marker reference
  - peak search left 6-77
  - peak search right 6-78
  - peak tracking 6-76
  - plotting main marker 10-26
  - position by display point 6-81
  - SEE ALSO sideband markers
  - to highest peak 6-75
  - X-axis position 6-81
- marker functions
  - damping 6-67
  - determining band power 6-66
  - determining sideband power 6-66
  - for correlation analysis data 6-67
  - for frequency data 6-66
  - for frequency response data 6-67
  - for octave analysis data 6-68
  - for order analysis data 6-68
  - for unfiltered time data 6-68
  - gain crossover 6-67
  - gain margin 6-67
  - maximum overshoot 6-67
  - phase crossover power 6-67
  - phase margin 6-67
  - resonant frequency 6-67
  - result 6-71
  - rise time 6-67
  - selecting 6-66
  - selecting a waterfall slice 6-132 - 6-133
  - selecting a waterfall trace 6-136 - 6-137
  - settling time 6-67
  - start value 6-52
  - steady state level 6-67
  - stop value 6-54
  - time delay 6-67
- marker reference
  - absolute position 6-79
  - assigning plotter pens 10-18
  - plotting 10-26
  - position relative to main marker 6-79, 6-91, 6-93
  - X-axis position 6-84
  - Y-axis position 6-85
- mass storage
  - SEE disk
- math
  - defining constants 6-94
  - defining functions 6-97
  - displaying constants 6-94, 6-99
  - displaying functions 6-99
  - enabling 6-100
  - loading definitions from disk 15-15
  - storing definitions to disk 15-31
  - transferring definitions via GPIB 6-95
- MAV bit 1-15
- measurement
  - data source 18-26
  - four-channel 12-9
  - one-channel 12-9
  - pausing 11-2
  - reference channel 18-58
  - restarting 11-2
  - results 6-23
  - starting 11-3
  - two-channel 12-9
- measurement data
  - SEE ALSO calculate data
  - displaying 6-23
  - Measuring bit 1-23
  - SEE ALSO trace data
  - types 23-3
  - valid for instrument mode 6-27
- measurement mode
  - SEE instrument mode
- memory
  - allocating for time capture 18-74
  - available 14-6
  - catalog 14-2
  - deleting 14-4 - 14-5
  - determining usage 14-3
  - purging 14-4 - 14-5
  - usage table 8-17
- message
  - SEE ALSO <MI>P-IB Programmer's Guide<D>
  - syntax 2-5 - 2-11
  - termination 2-6
- Message Available bit 1-15
- mnemonic, echoing 8-8
- most active trace 6-3, 6-56

## INDEX (Continued)

### N

naming plot/print file 15-25  
negative transition register  
  Device State 20-5  
  effect by preset 20-12  
  Limit Fail 20-19  
  Operation Status 20-10  
  Questionable Status 20-21  
  Questionable Voltage 20-26  
new line character (NL) 2-6  
normalized variance 6-26  
Nyquist diagram 6-30

### O

octave analysis instrument mode 13-2 - 13-3  
octave measurement  
  1/1 octave band 18-38  
  1/12 octave band 18-38  
  1/3 octave band 18-38  
  averaging 18-4, 18-17  
  equal confidence averaging 18-2  
  impulse detection 18-5  
  overall band 8-24 - 8-25  
  type 18-38  
offset marker  
  SEE marker reference  
1/1 octave band measurement 18-38  
\*OPC 3-7  
\*OPC? 3-7  
Operation Status register set  
  condition register 20-7  
  defined 1-22  
  enable register 20-8  
  event register 20-9  
  negative transition register 20-10  
  positive transition register 20-11  
\*OPT? 3-9  
option configuration  
  displaying 8-17  
  query 3-9  
orbit diagram 6-24  
order analysis instrument mode 13-2 - 13-3  
order resolution 18-52  
order spectrum measurement  
  SEE ALSO order tracking  
  selecting 18-57  
order tracking  
  composite power 6-26  
  displaying an order track 6-25  
  include DC bin 18-89  
  maximum RPM 18-54  
  minimum RPM 18-55  
  number of orders 18-51  
  resolution 18-52 - 18-53  
  RPM profile 6-25

  selecting 18-57  
  specifying order number 18-56  
  SEE ALSO tachometer  
output, selecting source 19-5  
overall band 8-24 - 8-25  
overlapped command  
  SEE ALSO <MI>P-IB Programmer's  
  Guide<D>  
  completion of 3-7  
  defined 3-10  
  processing 3-20  
overload rejection 18-59

### P

page-eject 10-8  
passing control 3-10  
pausing  
  a measurement 11-2  
  a program 17-12  
\*PCB 3-10  
PDF 6-25  
PEAK amplitude 6-111  
peak search  
  continuous 6-76  
  peak tracking 6-76  
  single 6-75  
  to left 6-77  
  to right 6-78  
pen assignments  
  SEE plotter  
phase  
  selecting coordinates unit 6-113  
  unwrapped 6-30  
  wrapped 6-29  
plotter  
  address 10-23  
  assigning annotation pens 10-9  
  assigning graticule pen 10-15  
  assigning marker pens 10-18  
  assigning P1 P2 values 10-19  
  assigning trace pens 10-14  
  defining line types 10-16 - 10-17  
  format 10-4  
  selecting 10-4  
  selecting default pens 10-2  
  setting P1 10-20  
  setting P2 10-22  
  speed setting 10-6  
plotting  
  entire screen 10-26  
  executing 10-7  
  graticule 10-26  
  grid 10-26  
  Hardcopy In Progress bit 1-23  
  label 10-10 - 10-11  
  main marker 10-26

- marker reference 10-26
- page-eject 10-8
- specifying a filename 15-25
- time stamp 10-13
- time stamp format 10-12
- title 10-25
- to a file 10-3
- to an GPIB device 10-3
- trace only 10-26
- polar diagram
  - angle of rotation 8-22
  - displaying 6-30
  - vector direction 8-21
- poles, number for curve fit 6-15
- positive transition register
  - Device State 20-6
  - effect by preset 20-12
  - Limit Fail 20-20
  - Operation Status 20-11
  - Questionable Status 20-22
  - Questionable Voltage 20-27
- power spectral density units 6-119
- power spectrum function 6-23
- power, setting 21-22
- Power-on Status Clear flag, setting 3-11
- preset
  - front panel 21-24
  - status 20-12
- preset states
  - SEE the individual commands
- preview averaging
  - accept 18-9
  - reject 18-10
  - setting 18-8
  - timed 18-11
  - Waiting for Accept/Reject bit 1-23
- printer
  - address 10-24
  - selecting 10-4
- printing
  - executing 10-7
  - format 10-4
  - Hardcopy in Progress bit 1-23
  - label 10-10 - 10-11
  - page-eject 10-8
  - specifying a filename 15-25
  - time stamp 10-13
  - time stamp format 10-12
  - title 10-25
  - to a file 10-3
  - to an GPIB device 10-3
- probability density function, displaying 6-25
- program
  - SEE Instrument BASIC
- program message
  - terminators 2-6
  - use 2-5
- \*PSC 3-11
- PSD units 6-119
- purging memory 14-4 - 14-5
- Q
- query
  - determining units 3-12
- Questionable Status register set
  - condition register 20-13
  - defined 1-18
  - enable register 20-14
  - event register 20-15
  - negative transition register 20-21
  - positive transition register 20-22
- Questionable Voltage register set
  - condition register 20-23
  - defined 1-19
  - enable register 20-24
  - event register 20-25
  - negative transition register 20-26
  - positive transition register 20-27
- queues 3-10
- quick verification 3-5
- R
- radian units 6-113
- range
  - autoranging 18-79, 18-81
  - manual selection 18-86
  - Ranging bit 1-23
- real
  - data transfer format 9-2
  - selecting coordinate system 6-29
- recalling
  - SEE loading
- reference level
  - SEE ALSO scaling
  - setting 8-39
  - tracking to input range 8-39
- register set
  - Device State 1-16
  - SEE ALSO Device State register set
  - Limit Fail 1-17
  - SEE ALSO Limit Fail register set
  - Operation Status 1-22
  - SEE ALSO Operation Status register set
  - Questionable Status 1-18
  - SEE ALSO Questionable Status register set
  - Questionable Voltage 1-19
  - SEE ALSO Questionable Voltage register set
  - Standard Event 1-20
  - SEE ALSO Standard Event register set
  - Status Byte 1-14
  - SEE ALSO Status Byte register set
  - summary 1-13, 1-25

## INDEX (Continued)

- transition of bits (condition register) 18-13
- User Status 1-24
- SEE ALSO User Status register set
- remove
  - SEE delete
- renaming files 15-23
- reset
  - cancelling \*OPC 3-8
  - executing 3-12
  - SEE ALSO preset
- resolution
  - bandwidth 18-40
  - frequency 18-40
  - number of display points 6-82
  - octave 18-38
  - order tracking 18-52 - 18-53
  - swept sine 18-34, 18-36 - 18-37
- response message 2-5
- restarting a measurement 11-2
- RMS amplitude 6-111
- rotational speed
  - SEE RPM
- RPM
  - arm starting value 5-5
  - arming 5-4
  - displaying RPM profile 6-25
  - indicator 8-13
  - maximum 18-54
  - minimum 18-55
  - step arming 5-3
- RS-232-C
  - SEE serial
- \*RST 3-12
- running a program 17-12
- S**
- sample programs
  - SEE example programs
- saving
  - SEE storing
- scaling
  - both traces 8-29, 8-34
  - bottom reference 8-36
  - center reference 8-37
  - dB reference 6-114 - 6-117
  - display 8-31 - 8-32
  - increment per decade 8-38
  - increment per vertical division 8-38
  - reference level 8-39
  - top reference 8-41
  - vertical autoscaling 8-35
  - X-axis 6-121, 6-123 - 6-127
  - X-axis spacing 8-33
  - Y-axis 8-42
- SCPI
  - compliance 1-26 - 1-38
  - confirmed commands 1-27
  - version 21-27
- screen
  - SEE display
- self-test 3-19
- sensitivity
  - SEE range
- serial
  - character format 21-10
  - character length 21-6
  - handshake (receiver) 21-7
  - handshake (transmitter) 21-11
  - pacing 21-7, 21-11
  - parity 21-8 - 21-9
- serial number 3-6
- Service Request enable register
  - setting 3-14
- Settling bit 1-23
- settling time 18-66
- sideband markers
  - sideband increment 6-88
  - carrier frequency 6-86
  - determining power 6-66
  - SEE ALSO marker
  - SEE ALSO marker functions
  - number of sidebands 6-87
- source
  - amplitude 19-10
  - amplitude ramp rate 19-17
  - SEE ALSO arbitrary source
  - burst length 19-2
  - DC offset 19-11
  - enabling 16-3
  - output level 19-10
  - repeat function 19-8
  - selecting output 19-5
  - sine frequency 19-3 - 19-4
- space character (WSP) 2-5
- span
  - frequency 18-40
  - full 18-43
- special syntactic elements 2-5
- square root power spectral density units 6-119
- \*SRE 3-14
- SRQ
  - SEE service request
- Standard Event register set
  - clearing 3-5
  - defined 1-20
  - reading 3-5
  - setting 3-4
- start frequency 18-46
- Start RPM Arming qualifier 5-4
- starting a measurement 11-3

- state
    - SEE instrument state
  - Status Byte register set
    - clearing 3-3
    - defined 1-14
    - reading 3-16
  - status registers 1-13 - 1-25
    - SEE ALSO <MI>P-IB Programmer's Guide<D>
  - \*STB? 3-16
  - step size
    - frequency setting 18-31
    - source amplitude 19-10
    - when frequency changes 18-48
  - stop frequency 18-49
  - stopping a program 17-12
  - storing
    - curve fit table (to disk) 15-26
    - curve fit table (via GPIB) 6-6
    - data tables (from disk) 15-28
    - instrument state (to disk) 15-34
    - instrument state (via GPIB) 21-25
    - lower limits (to disk) 15-29
    - lower limits (via GPIB) 6-39
    - math definitions (to disk) 15-31
    - math definitions (via GPIB) 6-95
    - programs (format) 15-33
    - programs (to disk) 15-32
    - programs (via GPIB) 17-3, 17-5
    - synthesis table (to disk) 15-35
    - synthesis table (via GPIB) 6-102
    - time capture file (large) 15-27
    - time capture file (to disk) 15-36
    - trace data (register) 23-2
    - trace data (via GPIB) 23-2
    - trace data files (to disk) 15-37
    - upper limit lines (to disk) 15-30
    - upper limit lines (via GPIB) 6-48
    - waterfall data (register) 23-5
    - waterfall data (via GPIB) 23-5
    - waterfall files (large) 15-27
    - waterfall files (to disk) 15-39
  - swept sine measurement
    - autolevel 19-12 - 19-13, 19-15 - 19-16
    - autolevel sensitivity 19-14
    - automatic resolution 18-34, 18-36 - 18-37
    - automatic sweep 18-62
    - frequency resolution 18-31
    - initial resolution 18-37
    - integration time 18-61
    - manual sweep 18-30, 18-62
    - normalized variance 6-26
    - resolution 18-34
    - settling time 18-66
    - source autolevel 19-9
    - source ramp rate 19-17
    - spacing between data points 18-65
    - sweep direction 18-60
  - synchronization 3-10
    - SEE ALSO <MI>P-IB Programmer's Guide<D>
  - syntax
    - conventions 2-6
    - message terminators 2-6
  - syntax descriptions 2-7
    - BLOCK 2-7
    - CHAR 2-7
    - CMDSTR 2-7
    - DEF\_BLOCK 2-8
    - EXPR 2-8
    - FILE 2-10
    - FILENAME 2-10
    - MMEMNAME 2-11
    - MSINAME 2-11
    - PROGRAM 2-11
    - STRING 2-11
  - synthesis
    - copy 6-101
    - creating 6-107
    - frequency scaling 6-105
    - gain constant 6-106
    - loading table from disk 15-18
    - loading values 6-102
    - scaling of the X-axis 6-108
    - selecting data register 6-104
    - starting 6-107
    - storing table to disk 15-35
    - SEE ALSO synthesis table
    - table format 6-110
    - time delay 6-109
  - synthesis table
    - displaying 8-17
    - loading from a file 15-18
    - storing to a file 15-35
    - transferring via GPIB 6-102
  - system controller 3-4, 3-8
- ## T
- tachometer
    - current RPM value 24-16
    - input range 24-15
    - number of pulses 24-14
    - RPM indicator 8-13
    - slope of input signal 24-17
    - trigger holdoff 24-12
    - trigger level 24-13
    - use in time capture 18-78
  - test
    - executing long confidence 22-3
    - results of long confidence 22-4
    - self-test 3-19
  - test log
    - clearing 22-2
    - displaying 8-17

## INDEX (Continued)

- 1/3 octave band measurement 18-38
- time capture
  - specifying channel 19-6
  - aborting 18-68
  - allocating memory 18-74
  - deleting 18-69
  - displaying 6-23, 8-16
  - executing 18-71
  - header information 8-17
  - length 18-72
  - loading file from disk 15-19
  - maximum RPM of tachometer 18-77
  - purging 14-5
  - size of buffer 18-72
  - source for measurement data 18-26
  - start (region) 18-75
  - stop (region) 18-76
  - storing file to disk 15-36
  - use of tachometer 18-78
- time capture mode 18-26
- time delay
  - curve fit 6-17
  - synthesis 6-109
- time instrument mode
  - SEE histogram instrument mode
- time record
  - displaying 6-24
  - length 18-67
  - number of points 18-27
  - overlap 18-63
- time stamp
  - enabling 10-13
  - format 10-12
- time step arming
  - SEE arm
- time, setting 21-26
- trace data
  - active trace 6-3
  - alias-protection 23-4
  - assign plotter pens 10-14
    - SEE ALSO calculate data
  - creating a label 8-27
  - default label 8-28
  - display format 8-7
  - displaying 8-17
  - loading file from disk 15-20
  - plotting 10-26
  - scaling 8-34
  - selecting 6-23
  - size of time record 18-27
  - storing file to disk 15-37
  - storing to data register 23-2
  - transferring via GPIB 23-2
  - types 23-3
  - X-axis data 23-7
  - X-axis unit 23-8
  - Y-axis amplitude unit 6-111
  - Y-axis units E-1 - E-6
  - Y-axis vertical unit 6-119
  - Z-axis data 23-9
  - Z-axis unit 23-10
- trace type
  - SEE coordinates
- transducer
  - direction 12-7
  - number of points 12-8
    - SEE ALSO transducer units
- transducer units
  - converting coordinates 6-118
  - enabling 18-84
  - labeling 18-82
  - selecting 18-85
  - sensitivity 18-83
- transferring data
  - coordinate-transformed data 6-20
  - curve fit table 6-6
  - data tables 6-59, 6-61
    - SEE ALSO example programs
  - instrument state 21-25
  - lower limit lines 6-39
  - math definitions 6-95
  - programs 17-3, 17-5
  - synthesis table 6-102
  - to data register 23-2
  - to external controller 6-20, 6-59, 6-61, 6-138
  - upper limit lines 6-48
  - waterfall data 6-129, 23-5
  - X-axis values 6-138
- transition of condition register bits 18-13
- \*TRG 3-18
- trigger
  - SEE ALSO arm
  - \*TRG 3-18
  - automatic 24-8
  - bus 24-5, 24-8
  - delay in collecting data 24-10
  - delay values 24-11
  - exiting the idle state 11-3
  - external 24-2 - 24-4, 24-8
  - free run 24-8
  - initiating 11-2
  - input channels 24-8
  - level 24-6
  - post-trigger value 24-10
  - pre-trigger value 24-10
  - slope 24-7
  - source 24-8
    - SEE ALSO tachometer
  - tachometer level 24-13
  - types 24-8
  - Waiting for TRIG bit 1-23
- \*TST? 3-19
- 1/12 octave band measurement 18-38



## U

unfiltered time record, displaying 6-25

uniform window 18-90

## units

amplitude 6-111

dB reference 6-114 - 6-117

determining for Y-axis E-1 - E-6

engineering 18-82, 18-84

phase coordinates 6-113

transducer 6-118, 18-84 - 18-85

transducer sensitivity 18-83

user 18-82

vertical 6-119

X-axis 6-121, 6-123 - 6-127, 23-8

Y-axis amplitude E-5

Y-axis defaults E-2

Y-axis vertical E-4

Z-axis 23-10

units, determining 3-12

unwrapped phase 6-30

## uploading data

SEE ALSO example programs

SEE ALSO loading

SEE ALSO transferring data

## User Status register set

defined 1-24

enable register 20-28

event register 20-29 - 20-30

reading event register 20-29

## V

## variable

SEE ALSO Instrument BASIC

reading and writing 17-11, 17-13

variable frequency resolution 18-31

verification program 3-7

SEE ALSO quick verification

## vertical scaling

SEE scaling

volt units 6-119

volt} {\plain \super\fl 2} {\plain \fl units 6-119

## W

\*WAI 3-20

## waterfall data

loading file from disk 15-21

Loading Waterfall bit 1-23

purging 14-5

storing file to disk 15-39

transferring via GPIB 23-5

## waterfall display

copying a slice 6-131

displaying traces 8-44 - 8-45, 8-51

enabling hidden traces 8-47

height of trace box 8-46

masking traces 8-43

number of traces 6-128

saving a trace 6-135

selecting a slice 6-132 - 6-133

selecting a trace 6-136 - 6-137

selecting format 8-7

skewed 8-48 - 8-49

storing to data register 23-5

transferring 6-129

turning on 8-50

Z-axis data 23-9

Z-axis unit 23-10

## waterfall register

displaying 6-23

purging 14-5

storing data 23-5

## windowing function

exponential time constant 18-87

force window width 18-88

types 18-90

wrapped phase 6-29

WSP 2-5

## X

X-axis units 6-121, 6-123 - 6-127

## xdcr unit

SEE transducer units

## Y

## Y-axis

amplitude unit 6-111

dB reference 6-114 - 6-117

determining units E-1 - E-6

transducer units 6-118

vertical unit 6-119

## Z

## Z-axis

data 23-9

determining value 6-70

unit 23-10

zeros, number for curve fit 6-16



## Guide to Agilent 35670A Documentation

If you are thinking about...	And you want to...	Then read...
<ul style="list-style-type: none"> <li>◆ Unpacking and installing the Agilent 35670A</li> </ul>	<p>Install the Agilent 35670A Dynamic Signal Analyzer</p> <p>Do operation verification or performance verification tests</p>	<p><i>Agilent 35670A Installation and Verification Guide</i></p> <p><i>Agilent 35670A Installation and Verification Guide</i></p>
<ul style="list-style-type: none"> <li>◆ Getting started</li> </ul>	<p>Make your first measurements with your new analyzer</p> <p>Review measurement basics</p> <p>Learn what each key does</p>	<p><i>Agilent 35670A Quick Start Guide</i></p> <p><i>Agilent 35670A Operator's Guide</i></p> <p>Use the analyzer's [ <b>Help</b> ] key</p>
<ul style="list-style-type: none"> <li>◆ Making measurements</li> </ul>	<p>Learn how to make typical measurements with the Agilent 35670A</p> <p>Understand each of the analyzer's instrument modes</p>	<p><i>Agilent 35670A Operator's Guide</i></p> <p><i>Agilent 35670A Operator's Guide</i></p>
<ul style="list-style-type: none"> <li>◆ Creating automated measurements</li> </ul> <p>(Instrument Basic is Option 1C2)</p>	<p>Learn the Instrument Basic interface</p> <p>Record keystrokes for a particular measurement</p> <p>Program with Instrument Basic</p>	<p><i>Using Instrument Basic with the Agilent 35670A</i></p> <p><i>Agilent 35670A Quick Start Guide</i></p> <p><i>Instrument Basic User's Handbook</i></p>
<ul style="list-style-type: none"> <li>◆ Remote operation</li> </ul>	<p>Learn about the GPIB</p> <p>Learn how to program with GPIB</p> <p>Find specific GPIB commands</p>	<p><i>GPIB Programmer's Guide</i></p> <p><i>GPIB Programming with the Agilent 35670A</i></p> <p><i>Agilent 35670A GPIB Commands: Quick Reference</i></p>
<ul style="list-style-type: none"> <li>◆ Using analyzer data with a PC application</li> </ul>	<p>Display or plot analyzer data on or from a Personal Computer</p> <p>Transfer analyzer data to a PC software application format</p> <p>Transfer data from a PC software application format to the analyzer (for example, to load data into a data register)</p>	<p><i>Standard Data Format Utilities: User's Guide</i></p>
<ul style="list-style-type: none"> <li>◆ Servicing the analyzer</li> </ul>	<p>Adjust, troubleshoot, or repair the analyzer</p>	<p><i>Agilent 35670A Service Guide</i></p>

---

### *Need Assistance?*

If you need assistance, contact your nearest Agilent Technologies Sales and Service Office listed in the Agilent Catalog. You can also find a list of local service representatives on the Web at:

***<http://www.agilent.com/find/assist>*** or *contact your nearest regional office listed below.*

If you are contacting Agilent Technologies about a problem with your Agilent 35670 Dynamic Signal Analyzer, please provide the following information:

- Model number: Agilent 35670A
- Serial number:
- Options:
- Date the problem was first encountered:
- Circumstances in which the problem was encountered:
- Can you reproduce the problem?
- What effect does this problem have on you?

You may find the serial number and options from the front panel of your analyzer by executing the following:

Press **[System Utility]**, [more], [serial number].

Press **[System Utility]**, [options setup].

If you do not have access to the Internet, one of these centers can direct you to your nearest representative:

---

<b>United States</b>	Test and Measurement Call Center (800) 452-4844 (Toll free in US)
<b>Canada</b>	(905) 206-4725
<b>Europe</b>	(31 20) 547 9900
<b>Japan</b>	Measurement Assistance Center (81) 426 56 7832 (81) 426 56 7840 (FAX)
<b>Latin America</b>	(305) 267 4245 (305) 267 4288 (FAX)
<b>Australia/New Zealand</b>	1 800 629 485 (Australia) 0800 738 378 (New Zealand)
<b>Asia-Pacific</b>	(852) 2599 7777 (FAX) (852) 2506 9285

---