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 4600 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

Ground (symbol)

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
**Regulatory Markings**

<table>
<thead>
<tr>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="C-tick" /></td>
<td>The C-tick mark is a registered trademark of the Spectrum Management Agency of Australia. This signifies compliance with the Australian EMC Framework regulations under the terms of the Radio Communications Act of 1992.</td>
</tr>
<tr>
<td><img src="image" alt="CE" /></td>
<td>The CE mark is a registered trademark of the European Community.</td>
</tr>
<tr>
<td><img src="image" alt="ICES/NMB-001" /></td>
<td>ICES/NMB-001 indicates that this ISM device complies with the Canadian ICES-001.</td>
</tr>
<tr>
<td><img src="image" alt="Cet appareil ISM est conforme à la norme NMB-001 du Canada." /></td>
<td>Cet appareil ISM est conforme à la norme NMB-001 du Canada.</td>
</tr>
<tr>
<td><img src="image" alt="Contains one or more of the six hazardous substances above the maximum concentration value (MCV), 40 Year EPUP." /></td>
<td>Contains one or more of the six hazardous substances above the maximum concentration value (MCV), 40 Year EPUP.</td>
</tr>
<tr>
<td><img src="image" alt="The CSA mark is a registered trademark of the CSA-International." /></td>
<td>The CSA mark is a registered trademark of the CSA-International.</td>
</tr>
<tr>
<td><img src="image" alt="This instrument complies with the WEEE Directive (2002/96/EC) marketing requirement. The affixed product label indicates that you must not discard this electrical/electronic product in domestic household waste." /></td>
<td>This instrument complies with the WEEE Directive (2002/96/EC) marketing requirement. The affixed product label indicates that you must not discard this electrical/electronic product in domestic household waste.</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

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

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

## CHAPTER 2: Introduction to the Command Reference

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding the Right Command</td>
<td>2-3</td>
</tr>
<tr>
<td>Command Syntax in the Agilent 35670A</td>
<td>2-5</td>
</tr>
<tr>
<td>Special Syntactic Elements</td>
<td>2-5</td>
</tr>
<tr>
<td>Conventions</td>
<td>2-6</td>
</tr>
<tr>
<td>Syntax Descriptions</td>
<td>2-7</td>
</tr>
</tbody>
</table>
### CHAPTER 3: Common Commands

- **CAL?**query ............................................... 3-2
- **CLS**command ............................................. 3-3
- **ESE**command/query .......................................... 3-4
- **ESR?**query ............................................... 3-5
- **IDN?**query ............................................... 3-6
- **OPC**command/query .......................................... 3-7
- **OPT?**query ............................................... 3-9
- **PCB**command ............................................. 3-10
- **PSC**command/query .......................................... 3-11
- **RST**command ............................................. 3-12
- **SRE**command/query .......................................... 3-14
- **STB?**query ............................................... 3-16
- **TRG**command ............................................. 3-18
- **TST?**query ............................................... 3-19
- **WAI**command ............................................. 3-20

### CHAPTER 4: ABORt

- **ABORt**command ............................................ 4-2

### CHAPTER 5: ARM

- **ARM[:IMMediate]**command .................................. 5-2
- **ARM:RPM:INCRement**command/query .......................... 5-3
- **ARM:RPM:MODE**command/query ................................... 5-4
- **ARM:RPM:THReshold**command/query ............................ 5-5
- **ARM:SOURce**command/query ..................................... 5-6
- **ARM:TIMer**command/query ...................................... 5-8

### CHAPTER 6: CALCulate

- **CALCulate[1|2|3|4]:ACTive**command/query ..................... 6-3
- **CALCulate[1|2|3|4]:CFIT:ABORt**command ...................... 6-4
- **CALCulate[1|2|3|4]:CFIT:COPIY**command ...................... 6-5
- **CALCulate[1|2|3|4]:CFIT:DATA**command/query .................. 6-6
- **CALCulate[1|2|3|4]:CFIT:DESTination**command/query ............ 6-8
- **CALCulate[1|2|3|4]:CFIT:FREQuency[:AUTO]**command/query ........ 6-9
- **CALCulate[1|2|3|4]:CFIT:FREQuency:STARt**command/query ......... 6-10
- **CALCulate[1|2|3|4]:CFIT:FREQuency:STOP**command/query ........... 6-11
- **CALCulate[1|2|3|4]:CFIT:FSCale**command/query .................. 6-12
- **CALCulate[1|2|3|4]:CFIT[:IMMediate]**command ................... 6-13
- **CALCulate[1|2|3|4]:CFIT:ORDer:AUTO**command/query .............. 6-14
- **CALCulate[1|2|3|4]:CFIT:ORDer:POLes**command/query ............. 6-15
- **CALCulate[1|2|3|4]:CFIT:ORDer:ZERos**command/query ............... 6-16
- **CALCulate[1|2|3|4]:CFIT:TDELay**command/query .................. 6-17
- **CALCulate[1|2|3|4]:CFIT:WEIGht:AUTO**command/query .............. 6-18
- **CALCulate[1|2|3|4]:CFIT:WEIGht:REGister**command/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]:FEED**command/query ....................... 6-23
- **CALCulate[1|2|3|4]:FORMat**command/query ..................... 6-29
- **CALCulate[1|2|3|4]:GDAPerture:APERture**command/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:CL:Ear[:IMMediate]**command .......... 6-35
- **CALCulate[1|2|3|4]:LIMit:LOWer:MOVE:Y**command .................. 6-36
- **CALCulate[1|2|3|4]:LIMit:LOWer:REPort[:DATA]?**query ............. 6-37
- **CALCulate[1|2|3|4]:LIMit:LOWer:REPort:YDAta?**query. ............... 6-38
- **CALCulate[1|2|3|4]:LIMit:LOWer:SEGMENT**command/query .......... 6-39
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: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]:WATerfall:COUNtcommand/query .......................... 6-128
CALCulate[1|2|3|4]:WATerfall[:DATA]?query .................................. 6-129
CALCulate[1|2|3|4]:WATerfall:SLICe:COPYcommand .......................... 6-130
CALCulate[1|2|3|4]:WATerfall:SLICe:SELectcommand/query ..................... 6-131
CALCulate[1|2|3|4]:WATerfall:SLICe:SELect:POINtcommand/query ............... 6-132
CALCulate[1|2|3|4]:WATerfall:TRACe:COPYcommand .......................... 6-133
CALCulate[1|2|3|4]:WATerfall:TRACe:SELectcommand/query ..................... 6-134
CALCulate[1|2|3|4]:WATerfall:TRACe:SELect:POINtcommand/query ............... 6-135
CALCulate[1|2|3|4]:X:DATA?query ........................................ 6-136
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]command/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:CLOCKwisecommand/query ......................... 8-21
DISPlay[:WINDow[1|2|3|4]]:Polar:ROTationcommand/query ....................... 8-22
DISPlay[:WINDow[1|2|3|4]]:TRACe:POWER[:STATe]command/query ............. 8-23
DISPlay[:WINDow[1|2|3|4]]:TRACe:POWer[:STATe]command/query ............... 8-24
DISPlay[:WINDow[1|2|3|4]]:TRACe:GRATicule:GRID[:STATe]command/query ... 8-25
DISPlay[:WINDow[1|2|3|4]]:TRACe:VOLtagecommand/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 ......................................... 8-40
DISPlay[:WINDow[1|2|3|4]]:TRACe:Y[:SCALe]:SPACingcommand/query .................................... 8-41
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:HEIGHTcommand/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:STATecommand/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[:IMMediate]command .......................................................................... 10-6
HCOPY:ITEM:FFEed:STATecommand/query ..................................................... 10-7
HCOPY:ITEM:LABel:COLORcommand/query ...................................................... 10-8
HCOPY:ITEM:LABel:STATecommand/query ......................................................... 10-9
HCOPY:ITEM:LABel:TEXTcommand/query ......................................................... 10-10
HCOPY:ITEM:TDSStamp:FORMatcommand/query ............................................ 10-11
HCOPY:ITEM:TDSStamp:STATecommand/query ................................................ 10-12
HCOPY:ITEM:TDSStamp: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:LYTPeommand/query .................. 10-16
HCOPY:ITEM[:WINDow[1|2|3|4]]:TRACe:LYTPeommand/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:LLEFTcommand/query ................................ 10-20
HCOPY:PAGE:DIMensions:USER:URIGHTcommand/query ................................ 10-21
HCOPY:PLOT:ADDResscommand/query ............................................................. 10-22
HCOPY:PRINT:ADDResscommand/query ........................................................... 10-23
HCOPY:TITLE[1|2]command/query ................................................................ 10-24
HCOPY:SOURcecommand/query ...................................................................... 10-25
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:DELetecommand ..................................................... 15-4
MMEMory:DISK:ADDResscommand/query ..................................... 15-5
MMEMory:DISK:UNITcommand/query ........................................... 15-6
MMEMory:FSYstem?query ....................................................... 15-7
MMEMory:INITializecommand .................................................. 15-8
MMEMory:LOAD:CFITcommand ................................................... 15-9
MMEMory:LOAD:CONTinuecommand/query .................................... 15-10
MMEMory:LOAD:MATHcommand ................................................. 15-14
MMEMory:LOAD:PROGramcommand ............................................ 15-15
MMEMory:LOAD:STATEcommand ................................................. 15-16
MMEMory:LOAD:SYNThesiscommand ........................................... 15-17
MMEMory:LOAD:TCAPturecommand ............................................. 15-18
MMEMory:LOAD:TRACecommand ................................................. 15-19
MMEMory:LOAD:WATerfallcommand .......................................... 15-20
MMEMory:MDIRectorycommand ................................................. 15-21
MMEMory:MOVEcommand ...................................................... 15-22
MMEMory:MSIScommand/query .................................................. 15-23
MMEMory:NAMEcommand/query ................................................. 15-24
MMEMory:STOR:CFITcommand ................................................... 15-25
MMEMory:STOR:CONTinuecommand/query .................................... 15-26
MMEMory:STOR:MATHcommand .................................................. 15-30
MMEMory:STOR:PROGramcommand ............................................. 15-31
MMEMory:STOR:PROGram:FORMatcommand/query ........................... 15-32
MMEMory:STOR:STATEcommand ................................................... 15-33
MMEMory:STOR:SYNThesiscommand ............................................ 15-34
MMEMory:STOR: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[:SELECTed]:LABelcommand/query ............................. 17-4
PROGram[:SELECTed]:DELETE:ALLcommand .................................. 17-5
PROGram[:SELECTed]:DELETE[:SELECTed]command ............................ 17-6
PROGram[:SELECTed]:LABelcommand/query ............................ 17-7
PROGram[:SELECTed]:MALlocatecommand/query ............................ 17-8
PROGram[:SELECTed]:NAMEcommand/query .............................. 17-9
PROGram[:SELECTed]:NUMBERcommand/query ............................ 17-10
PROGram[:SELECTed]:NUMBerclear/order ................................. 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:RESult:RATEcommand/query ............................ 18-6
[SENSe:]AVERage:RESult[: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:FOOTer:STARt?query ............................ 18-22
[SENSe:]DATA:HEADER:FOOTer:FOOTer:STOP?query .......................... 18-23
[SENSe:]DATA:HEADER:POINtsquery ...................................... 18-24
[SENSe:]DATA:RANGecommand/query ...................................... 18-25
[SENSe:]FEEDeorder/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[:INCReament]command/query .......................... 18-48
CHAPTER 19: SOURce

SOURce: BURSt command/query ................................................................. 19-2
SOURce: FREQuency[:CW] command/query .............................................. 19-3
SOURce: FREQuency: FIXed command/query ............................................. 19-4
SOURce: FUNCTION[:SHApe] command/query ............................................ 19-5
SOURce: USER: CAPTure command/query .................................................. 19-6
SOURce: USER: [:REGister] command/query ............................................ 19-7
SOURce: USER: REPeat command/query .................................................... 19-8
SOURce: VOLTage[:LEVEL]: AUTO command/query ..................................... 19-9
SOURce: VOLTage[:LEVEL]: [:IMMediate][:AMPLitude] command/query .......... 19-10
SOURce: VOLTage[:LEVEL]: [:IMMediate]: OFFSet command/query ................ 19-11
SOURce: VOLTage[:LEVEL]: REference command/query ................................ 19-12
SOURce: VOLTage[:LEVEL]: REference: CHAnnel command/query .................. 19-13
SOURce: VOLTage[:LEVEL]: REference: TOLerance command/query ................ 19-14
SOURce: VOLTage[:LEVEL]: LIMit[:AMPLitude] command/query ..................... 19-15
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTem:VERSion?query</td>
<td>21-27</td>
</tr>
<tr>
<td><strong>CHAPTER 22: TEST</strong></td>
<td></td>
</tr>
<tr>
<td>TEST:LOG:CLEarcommand</td>
<td>22-2</td>
</tr>
<tr>
<td>TEST:LONGcommand</td>
<td>22-3</td>
</tr>
<tr>
<td>TEST:LONG:RESult?query</td>
<td>22-4</td>
</tr>
<tr>
<td><strong>CHAPTER 23: TRACe</strong></td>
<td></td>
</tr>
<tr>
<td>TRACe[:DATA]command/query</td>
<td>23-2</td>
</tr>
<tr>
<td>TRACe:WATERfall[:DATA]command/query</td>
<td>23-5</td>
</tr>
<tr>
<td>TRACe:X[:DATA]?query</td>
<td>23-7</td>
</tr>
<tr>
<td>TRACe:X:UNIT?query</td>
<td>23-8</td>
</tr>
<tr>
<td>TRACe:Z[:DATA]query</td>
<td>23-9</td>
</tr>
<tr>
<td>TRACe:Z:UNIT?query</td>
<td>23-10</td>
</tr>
<tr>
<td><strong>CHAPTER 24: TRIGger</strong></td>
<td></td>
</tr>
<tr>
<td>TRIGger:EXTernal:FILTER[:LPAS][:STATe]command/query</td>
<td>24-2</td>
</tr>
<tr>
<td>TRIGger:EXTernal:LEVelcommand/query</td>
<td>24-3</td>
</tr>
<tr>
<td>TRIGger:EXTernal:RANGecommand/query</td>
<td>24-4</td>
</tr>
<tr>
<td>TRIGger:[IMMediate]command</td>
<td>24-5</td>
</tr>
<tr>
<td>TRIGger:LEVelcommand/query</td>
<td>24-6</td>
</tr>
<tr>
<td>TRIGger:SLOPecommand/query</td>
<td>24-7</td>
</tr>
<tr>
<td>TRIGger:SOURcecommand/query</td>
<td>24-8</td>
</tr>
<tr>
<td>TRIGger:STARt[1</td>
<td>2</td>
</tr>
<tr>
<td>TRIGger:TACHometer:HOLDoffcommand/query</td>
<td>24-12</td>
</tr>
<tr>
<td>TRIGger:TACHometer:LEVelcommand/query</td>
<td>24-13</td>
</tr>
<tr>
<td>TRIGger:TACHometer:PCOuntcommand/query</td>
<td>24-14</td>
</tr>
<tr>
<td>TRIGger:TACHometer:RANGecommand/query</td>
<td>24-15</td>
</tr>
<tr>
<td>TRIGger:TACHometer[:RPM]?query</td>
<td>24-16</td>
</tr>
<tr>
<td>TRIGger:TACHometer:SLOPecommand/query</td>
<td>24-17</td>
</tr>
</tbody>
</table>

**APPENDIX A: Agilent 35670A Command Summary**

Command List | A-1

**APPENDIX B: Cross-Reference from Front-Panel Keys to GPIB Commands**

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
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.
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.
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.

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.
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 CONTROLLER** ] 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.

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; "RST"
50   OUTPUT @Agilent35670A; "SENS:FREQ:SPAN:FULL"
60   OUTPUT @Agilent35670A; "SENS:FREQ:SPAN?"
70   ENTER @Agilent35670A; A
80   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 CONTROLLR] 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[1234]]:TRACe:GRATicule[:IMMediate]
- HCOPY:ITEM[:WINDOW[1234]]:TRACe[:IMMediate]
- HCOPY:ITEM[:WINDOW[1234]]:TRACe:MARKer[:IMMediate]
- HCOPY:ITEM[:WINDOW[1234]]: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:

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

* only when an Instrument BASIC program is running

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

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:EXPLlicit: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.
The Agilent 35670A’s Status Registers

**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.

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

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

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

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.

![Diagram of Questionable Voltage Register Set](image)

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.
The Standard Event register set monitors GPIB errors and synchronization conditions. See figure 1-11.

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.
The Agilent 35670A’s Status Registers

**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.
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.

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.
The Agilent 35670A’s Status Registers

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
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]:EXPonential
[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:ADDRes
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]:SBI
SYSTem:COMMunicate:SERial:TRANsmi
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 PHPGl
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:COPY
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:COPY[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
SCPI Compliance Information

CALCulate[1|2|3|4]:WATERfall:SLICe:COPY
CALCulate[1|2|3|4]:WATERfall:SLICe:SELect
CALCulate[1|2|3|4]:WATERfall:SLICe:SELect:POINt
CALCulate[1|2|3|4]:WATERfall:TRACE:COPY
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:TDSTimestamp: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:LLEf
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:DATABle: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:TCAPlture
MMEMory:LOAD:WATerfall
MMEMory:MDIRectory
MMEMory:STORe:CFIT
MMEMory:STORe:CONTinue
MMEMory:STORe:CONTinue?
MMEMory:STORe:DATABle: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:TCAPlture
MMEMory:STORe:WATerfall

OUTPut:FILTer[:LPASts][:STATe]

PROGram:EDIT:ENABle
PROGram:EXPLicit:LABel

1-35
PROGram[:SELection]:LABel

[SENSE:]AVERage:CONFidence
[SENSE:]AVERage:HOLD
[SENSE:]AVERage:IM Pulse
[SENSE:]AVERage:RES ult:RATE
[SENSE:]AVERage:RES ult[: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:BLOCksiz e
[SENSE:]FREQuency:RESolution:AUTO:MI Nimum
[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:DELet e
[SENSE:]TCAPture[:IMMediate]
[SENSE:]TCAPture:LENGth
[SENSE:]TCAPture:MAILocate
[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: VOL Tage[: L EVel]: AUTO
SOURce: VOL Tage[: L EVel][: IMMediate]: OFFSet
SOURce: VOL Tage[: L EVel]: REFerence
SOURce: VOL Tage[: L EVel]: REFerence: CHAnnel
SOURce: VOL Tage[: L EVel]: REFerence: TOLerance
SOURce: VOL Tage: LIMit: INPut
STATus: QUEStionable: LIMit: CONDition
STATus: QUEStionable: LIMit: ENABLe
STATus: QUEStionable: LIMit[: EVENt]
STATus: QUEStionable: LIMit: NTR ansition
STATus: QUEStionable: LIMit: PTR ansition
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: NTR ansition
STATus: DEVice: PTR ansition

TEST: LOG: CLEar
TEST: LONG
TEST: LONG: RES ult

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

TRIGger: EXTernal: F I LTer[: LPAS][: STATe]
TRIGger: EXTernal: L EVel
TRIGger: EXTernal: R ANGe
TRIGger: L EVel: 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.

---

**FORMAT[:DATA]**

Command Syntax: `FORMAT[:DATA] ASCii|REAL, [<number>|<bound>]`

Example Statements:
- `OUTPUT 711;"FORMAT:DATA ASCii, 8"`
- `OUTPUT 711;"FORM REAL, 64"`

Query Syntax: `FORMAT[:DATA]`?

Return Format: `ASC,NR1`

Attribute Summary:
- Option: not applicable
- Synchronization Required: no
- Preset State: ASC 12
- SCPI Compliance: confirmed

Description:

`FORMAT:DATA` only affects data transfers initiated by the following commands:
- `CALC:DATA?`
- `CALC:CFIT:DATA?`
- `CALC:LIM:LOW:REP?`
- `CALC:LIM:LOW:SEG?`

---

2-1. Sample Command Reference Page
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.
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.
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
XFRequency:POWer [1|2|3|4] selects power spectrum
XFRequency:POWer:COHerence [1,2|1,3|1,4|3,4] selects coherence
XFRequency:POWer:CROSs [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|2|3|4|5] selects order track
XRPM:PROFile selects RPM profile
XTIME:CORRelation [1|2|3|4] selects auto correlation
XTIME:CORRelation:CROSs [1,2|1,3|1,4|3,4] selects cross correlation
XTIME:VOLTage:CDF [1|2|3|4] selects cumulative density function
XTIME:VOLTage:PDF [1|2|3|4] selects probability density function
XTIME:VOLTage:WINDow [1|2|3|4] selects windowed time
XVOLtage:VOLTage [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>]...<NL>
<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>>)...<NL>
<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 θ
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 θ
+    Add
-    Subtract
*    Multiply
/    Divide

**Operands:**
D1|D2|...D 8 Contents of data registers
F1|F2|...F 5 Contents of function registers
K1|K2|...K 5 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)
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)

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.

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>]]:”

<MMEMNAME>
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””.

Command Syntax in the Agilent 35670A
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.
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^{|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_revision>

**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.
Common Commands

**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.

*OPT?

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.
*PSC

Sets the state of the Power-on Status Clear flag.

Command Syntax:

```
*PSC <number>|<bound>
```

- `<number>` ::= a real number (NRf data)
  - limits: -32767:32767
- `<bound>` ::= MAX|MIN

Example Statements:

```
OUTPUT 711; "*PSC 1"
OUTPUT 711; "*ESE 1; *SRE 16; *PSC 0" ! Generate SRQ on
               ! powerup
```

Query Syntax:

```
*PSC?
```

Return Format:

```
Integer
```

Attribute Summary:

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

Description:

This command lets you specify whether or not the Service Request enable register and the Event Status enable register should be cleared (all bits reset to 0) at power-up.

The settings of the Service Request enable register and the Event Status enable register are saved in nonvolatile memory when you turn the analyzer off. These settings can be recalled when you turn the analyzer on, but only if the Power-on Status Clear (PSC) flag is reset to 0. When the PSC flag is set to 1, the two enable registers are cleared at power-up. Use *PSC to specify the state of the PSC flag.

The number last specified with *PSC 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.

If you want the analyzer to generate a service request at power-up, bit 7 of the Event Status enable register and bit 5 of the Service Request enable register must be set. This is only possible if the PSC flag is reset to 0.

The query returns the current state of the PSC flag.
**RST**

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.

**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 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.
Common Commands

*SRE

Sets bits in the Service Request enable register.

**Command Syntax:**

\[*SRE* <number>|<bound>*

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

\[\frac{2^{bit\_number}}{bit\_number}\]

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**

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?**

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
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.
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**

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: RPM: 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**

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.
The [SENSe:]AVERage:TCONEntrol REPeat command allows you to get a complete set of averages at each arming event during manual, RPM step and time step arming.

Figure 5-2. Recorded Time of Triggered Event
Referenced to Start Time
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
This subsystem contains commands that control the processing of measurement data. The block diagram figure 6-1 shows you how measurement data is processed.

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.

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

**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**

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

**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: CALCu late:CFIT:DATA <BLOCK>

<BLOCK> ::= see Description

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

Query Syntax: CALCu late[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
<point175> ::= 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
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:STARt command/query

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

Command Syntax:

```
CALCulate[1 | 2 | 3 | 4]:CFIT:FREQuency:STARt {<number>[<unit>]|<step>|<bound>
```

- `<number>` ::= a real number (NRf data)
  - limits: 0:114999.9023
- `<unit>` ::= [HZ]
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

Example Statements:

OUTPUT 711;"CALC4:CFIT:FREQ:START 10.2"

OUTPUT 711;"calc3:cfit:frequency:star 600"

Query Syntax:

```
CALCulate[1 | 2 | 3 | 4]:CFIT:FREQuency:STARt?
```

Return Format:

Real

Attribute Summary:

- Option: 1D3 Curve Fit/Synthesis
- Synchronization Required: no
- Preset State: +0.00
- 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: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; "CALC: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]**

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

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**

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: ORDER: ZERos
command/query

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

**Command Syntax:**

```
CALCulate[1|2|3|4]: CFIT: ORDER: ZERos <number>|<step>|<bound>
```

- `<number>` ::= a real number (NRf data)  
  limits: 0:20
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

**Example Statements:**

```
OUTPUT 711; "CALCULATE3: CFIT: ORDER: ZEROS 5"
OUTPUT 711; "CALC4: CFIT: ORDER: ZEROS 1"
```

**Query Syntax:**

```
CALCulate[1|2|3|4]: CFIT: ORDER: ZERos?
```

**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 zeros 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 zeros is used for the model. The number specified with this command represents the upper bounds for the iterative search. The number of zeros will not exceed the number specified by CALC:CFIT:ORD:ZER.

If CALC:CFIT:ORD:AUTO OFF is sent, the number of zeros 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;'"CALC: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:WEIGHT: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>
<thead>
<tr>
<th>Resolution</th>
<th>FFT Instrument Mode</th>
<th>Baseband</th>
<th>Zoom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of frequency points (frequency data)</td>
<td>Number of real time points (time data)</td>
<td>Number of frequency points (frequency data)</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>256</td>
<td>101</td>
</tr>
<tr>
<td>200</td>
<td>201</td>
<td>512</td>
<td>201</td>
</tr>
<tr>
<td>400</td>
<td>401</td>
<td>1024</td>
<td>401</td>
</tr>
<tr>
<td>800</td>
<td>801</td>
<td>2048</td>
<td>801</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Correlation Instrument Mode (no complex data)</th>
<th>Auto- and cross correlation (real data)</th>
<th>Time domain (real data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to T/2</td>
<td>-T/2 to T/2</td>
<td>-T/4 to T/4</td>
</tr>
<tr>
<td>100</td>
<td>128</td>
<td>256</td>
<td>128</td>
</tr>
<tr>
<td>200</td>
<td>256</td>
<td>512</td>
<td>256</td>
</tr>
<tr>
<td>400</td>
<td>512</td>
<td>1024</td>
<td>512</td>
</tr>
<tr>
<td>800</td>
<td>1024</td>
<td>2048</td>
<td>1024</td>
</tr>
</tbody>
</table>
CALCulate[1 | 2 | 3 | 4]:DATA:HEADer:POINts?

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:Point?"
```

**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

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:**

<table>
<thead>
<tr>
<th>Option</th>
<th>not applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Required</td>
<td>yes</td>
</tr>
<tr>
<td>Preset State</td>
<td>not applicable</td>
</tr>
<tr>
<td>SCPI Compliance</td>
<td>confirmed</td>
</tr>
</tbody>
</table>

**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]’.
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:RE Freelence 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:RE Freelence 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:RE Freelence 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 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:RE Freelence 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:RE Freelence 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).
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>
<thead>
<tr>
<th>Measurement Data</th>
<th>INSTRument:SELeect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>Auto correlation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Capture buffer</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Coherence</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Composite Power</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Correlation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Spectrum</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cumulative Density Function</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Register</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency Response</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Variance</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Orbit Diagram</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Order Track</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability Density Function</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RPM Profile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Unfiltered Time</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Measurement Data</th>
<th>CALC::FEED command</th>
<th>FFT</th>
<th>CORR</th>
<th>HIST</th>
<th>ORD</th>
<th>OCT</th>
<th>SINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall Register</td>
<td>[W1</td>
<td>W2</td>
<td>W3</td>
<td>W4</td>
<td>W5</td>
<td>W6</td>
<td>W7</td>
</tr>
<tr>
<td>Windowed Time</td>
<td>XTIM::VOLT::WIND [1</td>
<td>2</td>
<td>3</td>
<td>4]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
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.
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.

\[ t_g = -\frac{d\phi}{d\omega} \]

Where
\[ \phi = \text{phase response in radians} \]
\[ \omega = \text{frequency in radians/second} \]

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.
CALCulate

**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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Aperture Step Size (in percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.0 PCT</td>
</tr>
<tr>
<td>200</td>
<td>0.5 PCT</td>
</tr>
<tr>
<td>400</td>
<td>0.25 PCT</td>
</tr>
<tr>
<td>800</td>
<td>0.125 PCT</td>
</tr>
</tbody>
</table>
CALCulate\{1 | 2 | 3 | 4\}:LIMit:BEEP[:STATe]

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:LIM: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

CALCulate[1 | 2 | 3 | 4]:LIMit: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:LIM:LOW:CLE"
OUTPUT 711;"calculate:lim:lower:cle:im"
```

**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**

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]?

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:limit:lower:report:Data?"
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, (FORM at ASCII) `<DEF_BLOCK>` takes the following form:

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

When data is binary-encoded, (FORM at REAL) `<DEF_BLOCK>` takes the following form:

```plaintext
<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.

```plaintext
<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:SEGMen

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

**Command Syntax:**

```
CALCulate[1|2|3|4]:LIMit:LOWer:SEGMen <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"
```

**Query Syntax:**

```
CALCulate[1|2|3|4]:LIMit:LOWer:SEGMen?
```

**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:SEGMent:CLEar** command

Deletes a segment from the lower limit line. Limit lines; deleting lower segment

**Command Syntax:**
```
CALCulate [1|2|3|4]:LIMit:LOWer:SEGMent:CLEar <number>|<bound>
```

- `<number>` ::= a real number (NRf data)
  - limits: -9.9e37:9.9e37
    - (depend upon displayed X-axis value)
- `<bound>` ::= MAX|MIN

**Example Statements:**
```
OUTPUT 711;"CALC:LIMIT:LOW:SEGMENT:CLE 1.99693e+37"
OUTPUT 711;"CALC:LIMIT:LOW:SEGMENT:CLE 2.15257e+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 a lower limit line, send CALC:LIM:LOW:CLE.
CALCulate

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.)


**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.
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).

CALCulate[1 | 2 | 3 | 4]:LIMIT:UPPer:REPort[:DATA]?

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, (FORM at ASCII) <DEF_BLOCK> takes the following form:

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

When data is binary-encoded, (FORM at 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, (FORM at ASCII) <DEF_BLOCK> takes the following form:

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

When data is binary-encoded, (FORM at 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:SEGMen command/query

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

**Command Syntax:**

```
CALCulate:LIMit:UPPer:SEGMen <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:up:segm 20000, -5, 3000, -5,
             80000, -2, 90000,-2"
```

**Query Syntax:**

```
CALCulate[1 | 2 | 3 | 4]:LIMit:UPPer:SEGMen?
```

**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 command

Converting the specified trace into an upper limit line.

**Command Syntax:**
```
CALCulate[1 | 2 | 3 | 4]; LIMIT:UPPer:TRAce[:IMMediate]
```

**Example Statements:**
```
OUTPUT 711; "CALCULATE3: LIMIT: UP: 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

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: \(-9.9\times10^37:9.9\times10^37\)
- `<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:START 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; "calclulate2: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:I mm"
```

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

Copies the data table from one trace to another trace.

**Command Syntax:**
CALCulate[1|2|3|4]:MARKer:DTABle:COPY[1|2|3|4]

**Example Statements:**
OUTPUT 711;":calc4:marker:dtab:copy1"
OUTPUT 711;"CALC:MARK:DTABLE:COPY3"

**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]?

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, (FORM at ASCII) <DEF_BLOCK> takes the following form:

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

When data is binary-encoded, (FORM at 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.
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]?

Returns the X values in the specified data table.

**Query Syntax:**
```
CALCulate[1 | 2 | 3 | 4]: MARKer: DTABle: X[ : DATA]?
```

**Example Statements:**
```
OUTPUT 711; "Calc: MARK: 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**

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

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

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:dtable:x:label '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:SELection:**

Selects the data table entry.

**Command Syntax:**

\[
\text{CALCulate}[1|2|3|4]:\text{MARKer}:\text{DTABle}:X:SELection[:POINt] <number>|<step>|<bound>
\]

- `<number>` ::= a real number (NRf data)
  - limits: 1:50
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

**Example Statements:**

- OUTPUT 711; "Calc:Mark:DTable:X:Select:POInt 23"
- OUTPUT 711; "CALC:MARKER:DTAB:X:SEL 32"

**Query Syntax:**

\[
\text{CALCulate}[1|2|3|4]:\text{MARKer}:\text{DTABle}:X:SELection[:POINt]?
\]

**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:INSe rt**
- **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|HPOWer|THD|BPOWer|BRMS|SPOWer|OVERshoot|RTIMe|STIMe|DTIMe|SSL|GMAR|PMAR|GCRoss|PCRoss|FREQ|DAMP|SINF|WEIG|TP|WP
```

**Example Statements:**
```
OUTPUT 711;"CALC:MARK:FUNC OVERSHOOT"
OUTPUT 711;"CALC: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|SINF|WEIG|TP|WP
```

**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.
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.
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]’)

---

CALCulate
<table>
<thead>
<tr>
<th>Marker Function</th>
<th>FFT</th>
<th>Swept Sine</th>
<th>Corr</th>
<th>Octave data</th>
<th>Order data</th>
<th>Hist data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Freq data</td>
<td>Time data</td>
<td>Freq data</td>
<td>Freq resp</td>
<td>Freq data</td>
<td></td>
</tr>
<tr>
<td>harmonic power</td>
<td>CALC:MARK:FUNC HPOW</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>total harmonic distortion</td>
<td>CALC:MARK:FUNC THD</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>band power</td>
<td>CALC:MARK:FUNC BPOW</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>square root band power</td>
<td>CALC:MARK:FUNC BRMS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>sideband power</td>
<td>CALC:MARK:FUNC SPOW</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>gain margin</td>
<td>CALC:MARK:FUNC GMAR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phase margin</td>
<td>CALC:MARK:FUNC PMAR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gain crossover</td>
<td>CALC:MARK:FUNC GCR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phase crossover</td>
<td>CALC:MARK:FUNC PCR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resonant frequency</td>
<td>CALC:MARK:FUNC FREQ</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>damping</td>
<td>CALC:MARK:FUNC DAMP</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>delay time</td>
<td>CALC:MARK:FUNC DTIM</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>maximum overshoot</td>
<td>CALC:MARK:FUNC OVER</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>rise time</td>
<td>CALC:MARK:FUNC RTIM</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>settling time</td>
<td>CALC:MARK:FUNC STIM</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>steady state</td>
<td>CALC:MARK:FUNC SSL</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>overall band</td>
<td>CALC:MARK:FUNC TPOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>weighted overall band</td>
<td>CALC:MARK:FUNC WPOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
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 \texttt{CALC:MARK:FUNC WEIGHT}. Send the \texttt{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 \texttt{CALC:MARK:FUNC SINFO}. Send the \texttt{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:FUNC: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.
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**

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

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

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:max: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:RIGHT"

**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.
CALCulate

CALCulate[1 | 2 | 3 | 4]:MARKer:POSITION command/query

Specifies the main marker’s independent axis position.

Command Syntax:

```
CALCulate[1 | 2 | 3 | 4]:MARKer:POSITION {<number>[<unit>] | <step>|<bound>}
```

- `<number>` ::= a real number (NRf data)
  - limits: $-9.9 \times 10^{37} : 9.9 \times 10^{37}$
- `<unit>` ::= [HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV]
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

Example Statements:

```
OUTPUT 711;"calc2:marker:pos 102400 HZ"
OUTPUT 711;"CALC:MARK:POS .013 S"
```

Query Syntax:

```
CALCulate[1 | 2 | 3 | 4]:MARKer:POSITION?
```

Return Format:

Real

Attribute Summary:

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

Description:

This command is identical to the CALC:MARK:X command with the exception of orbits measurement data (CALC:FEED ‘XVOL:VOLT [1,2|1,3|1,4]’) or when measurement data is displayed in a Nyquist diagram (CALC:FORM NYQ) or a polar diagram (CALC:FORM POL).

In **orbits** (CALC:FEED ‘XVOL:VOLT [1,2|1,3|1,4]’), the independent axis is labeled time (T). The main marker displays the values for the three axis; X for the X-axis value, Y for the Y-axis value and T for the independent axis (in seconds). This command moves the main marker to a position along the T-axis.

In a **Nyquist diagram** (CALC:FORM NYQ) and a polar diagram (CALC:FORM POL), the X-axis is the real component of the measurement data and the Y-axis is the imaginary component. The independent axis is determined by the instrument mode. (Depending upon the instrument mode, the independent axis may be in terms of frequency, time, volts, RPM or orders.) This command moves the main marker to a position along the independent axis.

**Note**

For measurement data other than orbits, Nyquist or polar diagrams, this command behaves like the CALC:MARK:X[:ABSolute] command.
CALCulate[1|2|3|4]:MARKer:POSITION:POINT command/query

Moves the main marker to a specific display point.

**Command Syntax:**
```
CALCulate[1|2|3|4]:MARKer:POSITION:POINT <number>|<step>|<bound>
```

- `<number>` ::= a real number (NRf data)
  - limits: 0:2047
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

**Example Statements:**
```
OUTPUT 711; "CALC4:MARKER:POS:POIN 558"
OUTPUT 711; "calculate2:mark:position:poin 49"
```

**Query Syntax:**
```
CALCulate[1|2|3|4]:MARKer:POSITION:POINT?
```

**Return Format:**
Integer

**Attribute Summary:**
- Option: not applicable
- Synchronization Required: no
- Reset State: +0
- SCPI Compliance: instrument-specific

**Description:**

This command specifies the main marker’s X-axis position 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 6-5.

In correlation analysis, the number of points displayed along the X-axis depends upon the resolution and the windowing function set with the [SENSe:]WINDow[:TYPE] command. See table 6-5.

In swept sine, the number of points displayed along the X-axis depends upon the number of lines of resolution and the setting for auto-resolution. See table 6-5.

In histogram analysis, the number of points is determined by the number of bins (set with the 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 order tracking measurement data and time capture data.

To specify the main marker’s position by a value along the X-axis, use the CALC:MARK:X command.
To specify the main marker’s position by a value along the independent axis, use the CALC:MARK:POS.
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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Baseband</th>
<th>Zoom (Start frequency $\sqrt{0}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of frequency points (frequency data)</td>
<td>Number of real time points (time data)</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>256</td>
</tr>
<tr>
<td>200</td>
<td>201</td>
<td>512</td>
</tr>
<tr>
<td>400</td>
<td>401</td>
<td>1024</td>
</tr>
<tr>
<td>800</td>
<td>801</td>
<td>2048</td>
</tr>
</tbody>
</table>

#### Correlation Instrument Mode

(no complex data)

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Auto- and cross correlation (real data)</th>
<th>Time domain (real data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to T/2</td>
<td>-T/2 to T/2</td>
</tr>
<tr>
<td>100</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>200</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>400</td>
<td>512</td>
<td>1024</td>
</tr>
<tr>
<td>800</td>
<td>1024</td>
<td>2048</td>
</tr>
</tbody>
</table>
Table 6-5. Number of displayed data points in variable resolution (continued)

Swept Sine Instrument Mode
(FREQ:RES:AUTO OFF)

<table>
<thead>
<tr>
<th>FREQ:RES (Number of measured points $^1$)</th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

$^1$ 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)

<table>
<thead>
<tr>
<th>FREQ:RES:AUTO:MIN (Number of measured points $^1$)</th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

$^1$ 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:REFe rence:X command/query

Specifies the marker reference X-axis position.

**Command Syntax:**

```
CALCulate[1|2|3|4]:MARKer:REFe rence: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:REFe rence: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:REFerence:Y command/query

Specifies the marker reference Y-axis position.

**Command Syntax:**

```
CALCulate{1|2|3|4}:MARKer:REFerence:Y {<number>[<unit>]|<step>|<bound>}
```

- `<number>` ::= a real number (NRf data)
  - limits: -9.9e37: 9.9e37
- `<unit>` ::= [DBVRMS|VPK|DBVPK|V|DBV|EU|DBEU|VRMS]
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

**Example Statements:**

```
OUTPUT 711; "calculate4:mark:ref:y 7.47833e+37"
OUTPUT 711; "Calc2:Marker:Ref:Y -8.7245e+37"
```

**Query Syntax:**

```
CALCulate{1|2|3|4}:MARKer:REFerence:Y?
```

**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 Y-axis position for the reference marker.

To specify the marker reference Y-axis position relative to the main marker’s position, use the CALC:MARK:Y:REL command.
CALCulate

**CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:CARRier** command/query

Specifies the carrier frequency used for sideband markers and calculations.

**Command Syntax:**

```plaintext
CALCulate[1|2|3|4]:MARKer:SIDeband:CARRier {<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:SID:CAR 2200 HZ"
- OUTPUT 711; "calculate3:marker:sideband:carrier 19.800e03 RPM"

**Query Syntax:**

```plaintext
CALCulate[1|2|3|4]:MARKer:SIDeband:CARRier?
```

**Return Format:**

Real

**Attribute Summary:**

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

**Description:**

To specify the number of sidebands, send the CALC:MARK:SID:COUN command.

To increment the carrier frequency value to the next largest point on the X-axis, send CALC:MARK:SID:CARR UP.

To decrement the carrier frequency value to the next smallest point on the X-axis, send CALC:MARK:SID:CARR DOWN.

You can also set the value with an expression. Send CALC:MARK:SID:CARR (CALC:MARK:X?) to set the carrier frequency value to the current X-axis marker value.

**Note**

When you shift the carrier frequency up or down, all the sideband markers shift up or down by the same amount.

The query returns the value of the carrier frequency (in X-axis units) currently used for sideband markers and calculations. The value is returned even if the sideband markers are not on.

To determine the X-axis units, send CALC:MARK:SID:CARR? UNIT.
**CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:COUNt**

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:COUN?) returns the number of sideband markers currently specified for the display. The value is returned even if the sideband markers are not on.
CALCulate

CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:INCRement command/query

Specifies the frequency increment (or delta) between sideband markers.

**Command Syntax:**

\[
\text{CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:INCRement} \{<\text{number}> \ [<\text{unit}>]\} <\text{step}> <\text{bound}>
\]

\(<\text{number}> \ ::= \text{a real number (NRf data)}\)
\n\(\text{limits: -9.9e37:9.9e37}\)
\n\(<\text{unit}> \ ::= [\text{HZ|ORD|CPM}]\)
\n\(<\text{step}> \ ::= \text{UP|DOWN}\)
\n\(<\text{bound}> \ ::= \text{MAX|MIN}\)

**Example Statements:**

```
OUTPUT 711; "calculate2:mark:sid:increment 600 hz"
OUTPUT 711; "calc:mark:sid:incr 1800 rpm"
```

**Query Syntax:**

CALCulate[1 | 2 | 3 | 4]:MARKer:SIDeband:INCRement?

**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 acceptable value, send CALC:MARK:SID:INCR UP.

To decrement the value to the next smallest acceptable value, send CALC:MARK:SID:INCR DOWN.

The query returns the current sideband increment value. To determine the X-axis units, send CALC:MARK:HARM:INCR? UNIT.
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

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:X:RELative**

Specifies the marker reference X-axis position relative to the main marker.

**Command Syntax:**

```
CALCulate[1|2|3|4]:MARKer:X:RELative {<number>[<unit>]|[<step>][<bound>]}
```

- `<number>` ::= a real number (NRf data)
  
  limits: -102400.0:102400.0

- `<unit>` ::= [HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV]

- `<step>` ::= UP|DOWN

- `<bound>` ::= MAX|MIN

**Example Statements:**

```
OUTPUT 711; "calculate:mark:x:relative 43013.2"
OUTPUT 711; "Calc:Marker:X:Rel -43742.4"
```

**Query Syntax:**

```
CALCulate[1|2|3|4]:MARKer:X: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 X-axis position relative to the main marker’s position. To query the units for the X-axis, send CALC:MARK:X:REL? UNIT.

To specify an absolute X-axis position for the marker reference, use the CALC:MARK:REF:X command.
CALCulate

**CALCulate[1 | 2 | 3 | 4]:MARKer:Y[:ABSolute]?**

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;"CALC:MARK:REF:Y 109.477"
OUTPUT 711;"CALC:MARK: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:CONStant[1|2|3|4|5] command/query

Defines the value of one of the constant registers.

**Command Syntax:**
```
CALCulate:MATH:CONStant[1|2|3|4|5] <real_part>
[,<imaginary_part>]
```

| <real_part> ::= <number>|<bound> |
| <number> ::= a real number (NRf data) |
| limits: -9.9e37:9.9e37 |

| <bound> ::= MAX|MIN |

| <imaginary_part> ::= <number>|<bound> |
| <number> ::= a real number (NRf data) |
| limits: -9.9e37:9.9e37 |

| <bound> ::= MAX|MIN |

**Example Statements:**
- OUTPUT 711; "CALCULATE2: MATH: CONS5 0.1151"
- OUTPUT 711; "calc: math: cons -1,1"

**Query Syntax:**
```
CALCulate[1|2|3|4]: MATH: CONStant[1|2|3|4|5]?
```

**Return Format:**
```
Real, Real
```

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

**Description:**

The analyzer assumes the first parameter is the real part of the constant. If the second parameter is used, an imaginary part is specified.

To use a constant in a math function, you must first load it into one of the analyzer’s five constant registers, 1 through 5. You can include the constant register’s name (K1|K2|K3|K4|K5) at the appropriate place in your function. Functions are defined with the CALC:MATH:EXPR command.

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**

This command ignores the trace specifier, CALC[1|2|3|4].
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> ::= 8 byte floating point number
(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[MATH:EXPRession[1|2|3|4|5]]

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 jω
  - 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
- $X \cdot j$ multiply by $j \cdot \omega$
- + 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)
  - 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

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
<Point175> ::= 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:sysnt: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 \( f/f_{\text{frequency scale}} \), 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 \( 1/f_{\text{frequency scale}} \).

**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

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:TDELay** command/query

Specifies a time delay value for the synthesis operation.

**Command Syntax:**

```
CALCulate[1|2|3|4]:SYNThesis: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:Synt:Tdel 63.7533"
OUTPUT 711; "CALCULATE:SYNT:TDELAY 68.8017"
```

**Query Syntax:**

```
CALCulate[1|2|3|4]:SYNThesis:TDELay?
```

**Return Format:**

Real

**Attribute Summary:**

- Option: 1D3 Curve Fit/Synthesis
- Synchronization Required: no
- Preset State: +0.0
- SCPI Compliance: instrument-specific

**Description:**

This command allows a time delay, that is a phase ramp, to be included in the synthesized response.

A positive delay produces a negative phase ramp.

**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

<table>
<thead>
<tr>
<th>Measurement Data</th>
<th>CALC:FEED command</th>
<th>CALC:UNIT:AMPL</th>
<th>RMS</th>
<th>PEAK</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto correlation</td>
<td>CALC:FEED 'XTIM:CORR'</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Capture buffer</td>
<td>CALC:FEED 'TCAP'</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Coherence</td>
<td>CALC:FEED 'XFR:POW:COH'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composite Power</td>
<td>CALC:FEED 'XFR:POW:COMP'</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>CALC:FEED 'XTIM:CORR:CROS'</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Cross Spectrum</td>
<td>CALC:FEED 'XFR:POW:CROS'</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cumulative Density Function</td>
<td>CALC:FEED 'XTIM:VOLT:CDF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency Response</td>
<td>CALC:FEED 'XFR:POW:RAT'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram</td>
<td>CALC:FEED 'XTIM:VOLT:HIST'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>CALC:FEED 'XFR:POW:LIN'</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Order Track</td>
<td>CALC:FEED 'XORD:TRACK'</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED 'XFR:POW'</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Probability Density Function</td>
<td>CALC:FEED 'XTIM:VOLT:PDF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPM Profile</td>
<td>CALC:FEED 'XRPM:PROF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>CALC:FEED 'XTIM:VOLT'</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Windowed Time</td>
<td>CALC:FEED 'XTIM:VOLT:WIND'</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
CALCulate[1|2|3|4]:UNIT:ANGLE command/query

Specifies the unit for phase coordinates.

**Command Syntax:**  
CALCulate[1|2|3|4]:UNIT:ANGLE DEGREE|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\[1\mid2\mid3\mid4\]:UNIT:DBReference \text{command/query}

Specifies the reference for dB magnitude trace coordinates.

**Command Syntax:**

\texttt{CALCulate[1\mid2\mid3\mid4]:UNIT DBReference \textless UNIT\textgreater}

\textless UNIT\textgreater :: = 'DBV' | 'DBV'

'DBM' | 'DBM'

'DBSLP' | 'DBSPL'

'DBUSER' | 'DBUSER'

**Example Statements:**

\texttt{OUTPUT 711; "CALC:UNIT:DBREFERENCE 'DBM'"}

\texttt{OUTPUT 711; "calc3:unit:dbr 'DBUSER'"}

**Query Syntax:**

\texttt{CALCulate[1\mid2\mid3\mid4]:UNIT:DBReference?}

**Return Format:**

"DBV" | "DBM" | "DBSPL" | "DBUSER"

**Attribute Summary:**

\texttt{Option: not applicable}

\texttt{Synchronization Required: yes}

\texttt{Preset State: "DBV"}

\texttt{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, \texttt{CALCulate[1\mid2\mid3\mid4]}, 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 \texttt{CALC:UNIT:DBR 'DBV'} to reference the dB magnitude to 1 volt. This is the default selection.

Send \texttt{CALC:UNIT:DBR 'DBM'} to reference the dB magnitude to 1 milliwatt. Use the \texttt{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 (\texttt{SENSE:VOLTAGE:RANGE:UNIT:USER:LABEL 'V'}).

Send \texttt{CALC:UNIT:DBR 'DBSPL'} to set the dB magnitude reference level to 20 µPa. An engineering unit of Pascals (Pa) must be applied to the data. See the \texttt{SENSE:VOLTAGE:RANGE:UNIT:USER} commands for additional information about enabling transducer units.

Send \texttt{CALC:UNIT:DBR 'DBUSER'} to set your own dB magnitude reference level. Use the \texttt{CALC:UNIT:DBR:USER:REFerence} command to specify the reference level. Use the \texttt{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 (\texttt{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 (Ω).

**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; "CALC:UNIT:DBR:IM 600 ohm"
- 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 Ω.

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: \[
\text{CALCulate}[1 | 2 | 3 | 4]:\text{UNIT}:\text{DBReference}:\text{USER}:\text{LABEL} \ ' <\text{NAME}>' \]

\(<\text{NAME}> ::= \text{ASCII} \text{ characters} \cdot 32 \text{ through} \ 126 \]
\(\text{maximum number of characters:} \ 5 \)

Example Statements:
OUTPUT 711; "CALC:UNIT:DBR:USER:LABEL 'g'"
OUTPUT 711; "calculate2:unit:dbreference:user:label 'm/s^2'"

Query Syntax:
\[
\text{CALCulate}[1 | 2 | 3 | 4]:\text{UNIT}:\text{DBReference}:\text{USER}:\text{LABEL}? \]

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/S², 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 volts², 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

<table>
<thead>
<tr>
<th>Measurement Data</th>
<th>CALC:FEED command (INST:SEL command)</th>
<th>CALC:UNIT:VOLT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite Power</td>
<td>CALC:FEED 'XFR:POW:COMP' (INST:SEL ORD)</td>
<td>X</td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>CALC:FEED 'XFR:POW:LIN' (INST:SEL FFT)</td>
<td>X</td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>CALC:FEED 'XFR:POW:LIN' (INST:SEL SINE)</td>
<td>X</td>
</tr>
<tr>
<td>Order Track</td>
<td>CALC:FEED 'XORD:TRACK' (INST:SEL ORD)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED 'XFR:POW' (INST:SEL FFT)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED 'XFR:POW' (INST:SEL OCT)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED 'XFR:POW' (INST:SEL ORD)</td>
<td>X</td>
</tr>
</tbody>
</table>
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.

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: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 'cpm'"
OUTPUT 711; "calc:unit:x:user:frequency:label 'rad/s'"
```

**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

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

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:LABEL '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**

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]?

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>).
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:COPY command

Copies the selected waterfall slice to the designated data register.

Command Syntax:

```
CALCulate[1 | 2 | 3 | 4]:WATerfall:SLICe:COPY
D1|D2|D3|D4|D5|D6|D7|D8
```

Example Statements:

```
OUTPUT 711; "calculate4:wat:slice:copy D1"
OUTPUT 711; "Calc3:Waterfall:Slice: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

CALCulate[1 | 2 | 3 | 4]:WATerfall:SLICe:SELect command/query

Selects the waterfall slice at the specified X-axis position.

**Command Syntax:**

\[
\text{CALCulate[1 | 2 | 3 | 4]:WATerfall:SLICe:SELect} \{ <\text{number}> \[ [<\text{unit}>] ] | <\text{step}> | <\text{bound}> \}
\]

- \( <\text{number}> :: a \text{ real \ number \ (NRf \ data)} \)
  - limits: \(-16382:1e6\)
- \( <\text{unit}> :: [ \text{HZ|S|ORD|COUNT|AVG|V|VPK|RPM|EU|REV} ] \)
- \( <\text{step}> :: \text{UP|DOWN} \)
- \( <\text{bound}> :: \text{MAX|MIN} \)

**Example Statements:**

```
OUTPUT 711; "CALC4:WAT:SLICE:SEL 880661"
OUTPUT 711; "calculate:wat:slic:select 757455"
```

**Query Syntax:**

\[
\text{CALCulate[1 | 2 | 3 | 4]:WATerfall:SLICe:SELect}?
\]

**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:Slice:Sel:Point 816"
OUTPUT 711;"CALC4:WATERFALL:SLIC:SELE: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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Baseband</th>
<th>Zoom (Start frequency ≠ 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of frequency points (frequency data)</td>
<td>Number of real time points (time data)</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>256</td>
</tr>
<tr>
<td>200</td>
<td>201</td>
<td>512</td>
</tr>
<tr>
<td>400</td>
<td>401</td>
<td>1024</td>
</tr>
<tr>
<td>800</td>
<td>801</td>
<td>2048</td>
</tr>
</tbody>
</table>

Swept Sine Instrument Mode
(FREQ:RES:AUTO ON)

<table>
<thead>
<tr>
<th></th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ:RES:AUTO:MIN (Number of measured points $^1$)</td>
<td>Number of displayed points</td>
</tr>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

$^1$ 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:COPY** command

Saves the selected trace to the specified data register.

**Command Syntax:**

```
CALCulate[1|2|3|4]:WATerfall:TRACE:COPY
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

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:

\[
\text{CALCulate[1|2|3|4]:WATerfall:TRACe:SELect:POINt } <\text{number}> \\
\mid <\text{step}>|<\text{bound}>
\]

\(<\text{number}> ::= a \text{ real number (NRf data)}\\n\mid \text{limits: 0:32767}\)

\(<\text{step}> ::= \text{UP|DOWN}\)

\(<\text{bound}> ::= \text{MAX|MIN}\)

Example Statements:

\text{OUTPUT 711;} \; \text{"CALC4:WATERFALL:TRAC:SELECT:POIN 4104"}

\text{OUTPUT 711;} \; \text{"calc4:waterfall:trac:select:poin 15879"}

Query Syntax:

\text{CALCulate[1|2|3|4]:WATerfall:TRACe:SELe}nt:POINt?

Return Format:

\text{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]?

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

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.
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.

**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**

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

Adjusts the intensity of the display.

**Command Syntax:**
```plaintext
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:**
```plaintext
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]**

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**

Selects a format for displaying trace data.

**Command Syntax:**
```
DISPLAY:FORM SING|ULOW|FBAC|SUBL|ULFB|QUAD
```

**Example Statements:**
```
OUTPUT 711; "DISPLAY:FORM ULFB"
OUTPUT 711; "display:form ULOWER"
```

**Query Syntax:**
```
DISPLAY:FORM?
```

**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.
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:PROGram:KEY:BOX**

Draws a box around softkey in an Instrument BASIC program.

**Command Syntax:**

\[
\text{DISPLAY:PROGRAM:KEY:BOX} \ <\text{key_number}> , \text{OFF|0|ON|1} \]

\[
<\text{key_number}> ::= <\text{number}>|<\text{bound}> \\
<\text{number}> ::= \text{a real number (NRf data)} \\
\text{limits: 0:8} \\
<\text{bound}> ::= \text{MAX|MIN} \\
\]

Example Statements:

\[
\text{OUTPUT 711;}^{*} ; ^{*} : ^{*} \text{disp:prog:key:box 1,ON}^{*} \\
\text{OUTPUT 711;}^{*} ; ^{*} \text{DISPLAY:PROGRAM:KEY:BOX 3,OFF}^{*} \\
\]

**Query Syntax:**

\[
\text{DISPLAY:PROGRAM:KEY:BOX?} \ <\text{key_number}> \\
\]

**Return Format:**

\[
\text{Integer, Integer} \\
\]

**Attribute Summary:**

\[
\text{Option: IC2 Instrument BASIC} \\
\text{Synchronization Required: no} \\
\text{Preset State: not applicable} \\
\text{SCPI Compliance: instrument-specific} \\
\]

**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:PROGRAM:KEY:BRACKET**

Draws a bracket around one or more softkeys in an Instrument BASIC program.

**Command Syntax:**
```plaintext
DISPLAY:PROGRAM: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:**
```plaintext
DISPLAY:PROGRAM: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:PROGرام[:MODE]**

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|UPPER|LOWER
```

**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:vector:buffer:state 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**

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; "DISPLAY: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**

Specifies what is displayed on the analyzer’s screen.

**Command Syntax:**

```
DISPlay:VIEW TRACe|MSTate|MMEMory|STABile|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]**

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.

**Command Syntax:**

```
DISPlay[:WINDow[1 | 2 | 3 | 4]]:DTABle[:STATe] OFF|0|ON|1
```

**Example Statements:**

```
OUTPUT 711; "DISPLAY:DTAB:STATE 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

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**

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:clock 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**: Wind[m|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:Pol: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.

8-24
**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[1234]** 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]

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[:WI NDow[1|2|3|4]]:TRACe:LABel command/query

Loads a label for the specified trace.

Command Syntax:  DISPlay[:WI NDow[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[:WI NDow[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]**

Turns on or off the analyzer’s default title for the specified trace.

**Command Syntax:**

```plaintext
DIS Play[: WINDow[1|2|3|4]]: TRACe: LABel: DEFault[: STATe] OFF|0|ON|1
```

**Example Statements:**

```plaintext
OUTPUT 711; "disp: wind4: trace: lab: default: stat OFF"
OUTPUT 711; "Disp: Trace: Lab: Default ON"
```

**Query Syntax:**

```plaintext
DIS Play[: 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**: ``DISPLAY[:WINDow[1|2|3|4]]::TRACe:X[:SCALe]:AUTO`` command/query

Scales the measurement data to fit the trace box. `$I.scaling;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**

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|LOGarithmetic
```

**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.
The 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

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:SCALE: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>|<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:pdvision
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 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.
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**

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**

Specifies scaling of the Y-axis for linear magnitude coordinate data.

**Command Syntax:**

```
DISPlay[:WINDow[1|2|3|4]]:TRACe:Y:SPACing Linear|Logarithmic
```

**Example Statements:**

```
OUTPUT 711;"Disp: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**  

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**

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:**

\[
\text{DISPlay[:WINDow[1|2|3|4]]:WATerfall:HEIGht \{<number>[<unit>]|<step>|<bound>\}}
\]

- `<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](image-url)
DISPlay[:WINDow[1|2|3|4]]:WATerfall:HIDDeN command/query

Turns on or off the removal of hidden waterfall traces.

Command Syntax:  DISPPlay[: 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**

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](image.png)
**DISPlay[:WINDow[1 | 2 | 3 | 4]]:WATerfall[:STATe]**

 Turns on or off the waterfall display for the specified trace box.

**Command Syntax:**

```
DISPlay[:WI NDow[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[:WI NDow[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 {<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; "DISp:Waterfall:Top 2.70387e+37"
OUTPUT 711; "DISP:WND3: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

This subsystem contains one command—FORMat:DATA. The command determines which data type and data encoding is used when your 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.

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**

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 a 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**

Specifies the format of the plot/print output.

**Command Syntax:**
```
HCOPy:DEV:LANGUAGE HPGL|PCL|PHPGL
```

**Example Statements:**
```
OUTPUT 711; "HCOPy:DEV:LANGUAGE hpgl"
OUTPUT 711; "HCOP:DEV:LANG PCL"
```

**Query Syntax:**
```
HCOPy:DEV: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**

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.
**HCOPYy**

**HCOPYy:DEVice:SPEed**  
(command/query)

Specifies the plotting speed for all plotting operations initiated by the analyzer.

**Command Syntax:**

```
HCOPYy: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:**

```
HCOPYy: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**

HCOPYy

HCOPYy:ITEM:FFEed:STATE command/query

Turns the page-eject feature on or off.

Command Syntax:  
HCOPYy:ITEM:FFEed:STATE OFF|0|ON|1

Example Statements:  
OUTPUT 711;"hcop:item:ffe:state ON"
OUTPUT 711;"Hcop:Item:Ffe:Stat OFF"

Query Syntax:  
HCOPYy: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**

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:**

**HCOPy:ITEM:LABEL:STATe**

Prints a label on the plot and print output.

**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 HCOP: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 HCOP:IMM command is executed.

To specify the label, use the HCOP:ITEM:LAB:TEXT command.

The color of the label is set with the HCOP:ITEM:LAB:COLOR command.

Send HCOP: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**

A line feed character divides the label into two lines. Time Stamp
**HCOPy:ITEM:LABel:TEXT**

Specifies a label for plot and print output.\textit{abeling:plot/print}

**Command Syntax:**

\texttt{HCOPy:ITEM:LABel:TEXT 'STRING'}

\texttt{<STRING> ::= maximum of 119 ASCII characters}

\texttt{59 characters per line}

Example Statements:

\texttt{OUTPUT 711;"hcop:ITEM:lab:te xt 'Test 1 Results'

OUTPUT 711;"HCOP:ITEM:LABEL:TEXT "LINE1"&CHR$(10)"LINE2"}

**Query Syntax:**

\texttt{HCOPy:ITEM:LABel:TEXT?}

**Return Format:**

\texttt{"<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 \texttt{<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:TITtle command.
**HCOPy:ITEM:TDSTamp:FORMat**

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:TDST:FORMAT FORMAT1"
OUTPUT 711;"HCOPY:ITEM:TDST:FORMAT FORMAT2"
```

**Query Syntax:**

```
HCOPY:ITEM:TDSTamp:FORMat?
```

**Return Format:**

```
FORM1 | FORM2 | FORM3 | FORM4 | FORM5
```

**Attribute Summary:**

<table>
<thead>
<tr>
<th>Option</th>
<th>not applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Required</td>
<td>no</td>
</tr>
<tr>
<td>Preset State</td>
<td>FORM5</td>
</tr>
<tr>
<td>SCPI Compliance</td>
<td>instrument-specific</td>
</tr>
</tbody>
</table>

**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.
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;"HCOPY:ITEM:TDSTAMP: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.
HCOPYy:ITEM[:WINDow[1 | 2 | 3 | 4]]:TRACe:COLor

Selects the pen used to plot the specified trace and annotation.

**Command Syntax:**

```
HCOPYy: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:TRACe:COL 0"
OUTPUT 711; ":hcopy:item:window:trace:color 6"
```

**Query Syntax:**

```
HCOPYy: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;"HCOP: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.
Selects the line type for the specified limit line.

**Command Syntax:**

```
HCOPy:ITEM[:WINDOW[1|2|3|4]]:TRACE:LIMIT:LTYPe
Solid|DASHed|DOTTed|STYLe<n>
```

<n> ::= a real number

limits: 0, 3-12

**Example Statements:**

```
OUTPUT 711; "HCOPy:ITEM:TRACE:LIMIT 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 HCOPy: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

Selects the line type for the specified trace.

**Command Syntax:**

```
HCOPy:ITEM[:WINDow[1|2|3|4]]:TRACe:LTYPe
SOLID|DASHed|DOTTed|STYLe<n>
```

\(<n> ::= a \text{ real number}\\
\text{limits: } 0, 3-12\)

**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:WIN3: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**

Specifies P1 and P2 values for a plotter.

**Command Syntax:**
```
HCOPy:PAGE:DI Mensions:AUTO OFF|0|ON|1
```

**Example Statements:**
```
OUTPUT 711;"HCOP:PAGE:DI M:AUTO ON"
OUTPUT 711;"hcopy:page:dimensions:auto OFF"
```

**Query Syntax:**
```
HCOPy:PAGE:DI Mensions: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:DI Mensions:USER:LLE F**

Specifies the lower left position (P1) of the plot area.

**Command Syntax:**

```
HCOPY: PAGE: DIM: USER: LLEF {<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; "HCOPY: PAGE: DIM: USER: LLEF 332, 4286"
OUTPUT 711; "HCOPY: PAGE: DIM: USER: LLEF 4743, 1195"
```

**Query Syntax:**

```
HCOPY: PAGE: DIM: USER: LLEF?
```

**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:DI Mensions: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:URIGHT 4743,7377"
OUTPUT 711; "hcopy:page:dimensions:user:uright 9155,4286"
```

Query Syntax:

```
HCOPY:PAGE:DI Mensions: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:URIGHT is set to +9155,+7377 after a reset (*RST).
HCOPy:PLOT:ADDRESS

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 HCOP:DESTination command is set to GPIB and the HCOP: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 HCOP: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]**

Specifies a label for plot and print output.

**Command Syntax:**

\[ \text{HCOPy:TITLE}[1|2|3|4] \ 'STRING' \]

\(<\text{STRING}>\) := maximum of 59 ASCII characters

**Example Statements:**

\(\text{OUTPUT 711;}\ "\text{hcopy:tit:lab:tex 'Test 1 Results'"} \)
\(\text{OUTPUT 711;} \ "\text{HCOP:TITLE:TExT2 'BEARING CHARACTERISTICS'"} \)

**Query Syntax:**

\(\text{HCOPy:TITLE[1|2]?}\)

**Return Format:**

\("<\text{STRING}>"\)

**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

HCOp y:SOURce

Selects the portion of the analyzer’s screen you want to plot.

Command Syntax:  

HCOp y:SOURce \{ALL|TRACe|MARKer|REFerence|GRID\}

Example Statements:  

OUTPUT 711;"HCOp y:SOUR GRID"
OUTPUT 711;"hcopy:sour MARKer"

Query Syntax:  

HCOp y: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 y: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 y:SOUR TRAC;IMM. Traces are plotted without grid lines, annotation or markers.

Send HCOp y: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 y: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 y:SOUR GRID;IMM to plot the graticule only for all displayed traces. The grid is plotted without the trace, markers or annotation.
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.
INITiate

**INITiate:CONTinuous**

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
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]

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

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:filter:awe:state 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]

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

**INPut[1 | 2 | 3 | 4]:LOW**  
*command/query*

Sets the specified channel’s input shield to float or to ground.

**Command Syntax:**  
\[ \text{INPut[1|2|3|4]:LOW GROUND|FLOAT} \]

**Example Statements:**  
OUTPUT 711; "Input3: Low GROUND"  
OUTPUT 711; "INP2: LOW FLOAT"

**Query Syntax:**  
\[ \text{INPut[1|2|3|4]:LOW?} \]

**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 Ω, send INP[1|2|3|4]:LOW GRO.

To float the analyzer’s input shield through 1 MΩ, 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 $\theta <$)
- 6 = P (tangential $\theta \phi$)
- 7 = TX
- 8 = TY
- 9 = TZ

The query response provides the current direction associated with the specified channel.
INPut

INPut[1 | 2 | 3 | 4]:RFerence:POINt command/query

Sets the number for the transducer point.

Command Syntax:

```
INPut[1 | 2 | 3 | 4]:RFerence: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:puin 11081"
OUTPUT 711;"INP:REF:POIN 12720"
```

Query Syntax:

```
INPut[1 | 2 | 3 | 4]:RFerence: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:**
\[ \text{INPut}[1|2|3|4][:STATe] \text{ OFF|0|ON|1} \]

**Example Statements:**
- OUTPUT 711;"inp2 OFF"
- OUTPUT 711;"Input:Stat ON"

**Query Syntax:**
\[ \text{INPut}[1|2|3|4][:STATe]? \]

**Return Format:**
\[ \text{Integer} \]

**Attribute Summary:**
- **Option:** not applicable
- **Synchronization Required:** no
- **Preset State:** Standard
  - INPut1 ON (+1)
  - INPut2 ON (+1)
  - INPut3 not available
  - INPut4 not available
- **Option AY6**
  - INPut1 ON (+1)
  - INPut2 ON (+1)
  - INPut3 OFF (+0)
  - 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 cannot 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

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)
- `<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]**

Selects one of the analyzer’s six major instrument modes.

**Command Syntax:**

```
INStrument[:SELe ct] FFT|OCTave|ORDer|SINE|HISTogram|CORRelation
```

**Example Statements:**

```
OUTPUT 711;":Inst SINE"
OUTPUT 711;"INST:SELECT FFT"
```

**Query Syntax:**

```
INStrument[:SELe ct]?
```

**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

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?

Returns information about memory usage allocated for a specific item.

Query Syntax:

MEMory:CATalog:NAME? TCAPture|WATerfall|WREGister|PRO-Gramp|RDISk|LIMits

Example Statements:

OUTPUT 711; "MEMORY: CAT: NAME? WREGISTER"
OUTPUT 711; "mem: cat: name? wat"

Return Format:

<NNAME>

<NNAME> ::= 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

MEMory:DELe:ALL  

Purges all allocated memory in the analyzer.

Command Syntax:  
MEMory:DELe: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]

Purges the memory allocated for a specific item.

**Command Syntax:**

```
MEMory:DELe[:NAME] TCAPture|WATerfall|WREGister|PRO-Gramp|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]?

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
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.
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 directory 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 characters 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.

<MEMNAME> ::= 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.
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

**MMEMory:DELe**te command

Deletes one file or the contents of an entire disk.

**Command Syntax:**

\[
\text{MMEMory:DELe} \ <\text{MMEMNAME}>
\]

\[
<\text{MMEMNAME}> ::= ' <disk> [ <dos_dir> ] [ <filename> ]'
\]

\[
<\text{disk}> ::= \text{NVRAM:|RAM:|INT:|EXT[,<select_code>[,<unit_number>]]}
\]

\[
<\text{select_code}> ::= \text{a real number (NRf data)}
\]

\[
\text{limits: 700:730}
\]

\[
<\text{unit_number}> ::= \text{a real number (NRf data)}
\]

\[
\text{limits: 0:3}
\]

\[
<\text{dos_dir}> ::= \text{directory_name}
\]

\[
\text{(DOS only)}
\]

\[
<\text{filename}> ::= \text{ASCII characters}
\]

\[
\text{(see beginning of chapter for <filename> restrictions)}
\]

**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 \(<\text{disk}>\). The select code and unit number specifiers are valid only with the \(<\text{EXT: disk specifier}>\).

You can use the “*” as a wildcard (for example, ‘*.DAT’).

**Note**

When accessing the external mass storage device (\(<\text{EXT:}>\)), 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**

**MMEMory:DISK:ADDRess**

*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**

**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:FSYStem?

Returns the type of file system for the default disk.

Query Syntax:

```
MMEMory:FSYStem?
```

Example Statements:

```
OUTPUT 711;":mmemory:fsys?"
OUTPUT 711;"Mmem:Fsystem?"
```

Return Format:

```
LIF|DOS
```

Attribute Summary:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>not applicable</td>
</tr>
<tr>
<td>Synchronization</td>
<td>Required: no</td>
</tr>
<tr>
<td>Preset State</td>
<td>not applicable</td>
</tr>
<tr>
<td>SCPI Compliance</td>
<td>instrument-specific</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Media</th>
<th>Format Option</th>
<th>Bytes/Sector</th>
<th>Sectors/Track</th>
<th>Tracks/Surface</th>
<th>Maximum Capacity (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-MByte</td>
<td>0</td>
<td>256</td>
<td>16</td>
<td>77</td>
<td>630,784</td>
</tr>
<tr>
<td></td>
<td>1*</td>
<td>256</td>
<td>16</td>
<td>77</td>
<td>630,784</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>512</td>
<td>9</td>
<td>77</td>
<td>709,632</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1,024</td>
<td>5</td>
<td>77</td>
<td>788,480</td>
</tr>
<tr>
<td></td>
<td>4**</td>
<td>256</td>
<td>16</td>
<td>77</td>
<td>270,336</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>512</td>
<td>9</td>
<td>80</td>
<td>737,280</td>
</tr>
<tr>
<td>2-MByte</td>
<td>0</td>
<td>256</td>
<td>32</td>
<td>77</td>
<td>1,261,568</td>
</tr>
<tr>
<td></td>
<td>1***</td>
<td>256</td>
<td>32</td>
<td>77</td>
<td>1,261,568</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>512</td>
<td>18</td>
<td>77</td>
<td>1,419,264</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1,024</td>
<td>10</td>
<td>77</td>
<td>1,576,960</td>
</tr>
<tr>
<td></td>
<td>4***</td>
<td>256</td>
<td>32</td>
<td>77</td>
<td>1,261,568</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>512</td>
<td>18</td>
<td>80</td>
<td>1,474,560</td>
</tr>
</tbody>
</table>

* 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 media.
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>{<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:TRACe1|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>[{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: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>\{<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: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>([{DOS_DIR}]]`
- `<MSIS>` ::= RAM: | NVRAM: | INT:
  - EXT[,<select_code>,<unit _number>]]:
- `<DOS_DIR>` ::= \<DIR_NAME>\ (DOS only)
- `<DIR_NAME>` ::= ASCII characters
- `<filename>` ::= ASCII characters (see beginning of chapter for `<filename>` restrictions)

**Example Statements:**

```
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**

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>[{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: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

**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> ::= <MSI>S[{DOS_DIR}]
<MSI>S ::= 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:program 'myprog'"
OUTPUT 711; "MMEM:LOAD:PROG 'INT:IBFILE'"
```

**Attribute Summary:**

<table>
<thead>
<tr>
<th>Option</th>
<th>not applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Required</td>
<td>no</td>
</tr>
<tr>
<td>Preset State:</td>
<td>not applicable</td>
</tr>
<tr>
<td>SCPI Compliance:</td>
<td>instrument-specific</td>
</tr>
</tbody>
</table>

**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>\n
- `<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

**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;"mmemory: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>{<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: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

MMEMory:LOAD:TRACe

Loads a trace into the analyzer from the specified disk.

Command Syntax:

```
MMEMory:LOAD:TRACe {D1|D2|D3|D4|D5|D6|D7|D8}, <FILENAME>[,<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:**

```
MMEMory:LOAD:WATerfall {W1|W2|W3|W4|W5|W6|W7|W8}, <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;"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 <OLDNAME>, <NEWNAME>
```

- `<OLDNAME>` ::= '<file>'
  - `<file>` ::= ASCII characters (see beginning of chapter for <filename> restrictions)
- `<NEWNAME>` ::= '<file>'
  - `<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 `<OLDNAME>` is the old file name; `<NEWNAME>` 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.
MEMory

MEMory:MSIS command/query

Specifies a default disk.

Command Syntax:

```
MEMory:MSIS <MSINAME>
```

```
<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:

```
MEMory:MSIS?
```

Return Format:

```
"NVRAM:" | "RAM:" | "INT:" | "EXT:";
```

Attribute Summary:

<table>
<thead>
<tr>
<th>Option</th>
<th>not applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Required</td>
<td>no</td>
</tr>
<tr>
<td>Preset State</td>
<td>saved in non-volatile memory</td>
</tr>
<tr>
<td>SCPI Compliance</td>
<td>confirmed</td>
</tr>
</tbody>
</table>

Description:

If you omit disk specifiers from MEMory 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 MEM:FSYstem? 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**

Specifies a filename for the output of a print or plot operation.

**Command Syntax:**

```
MMEMory:NAME <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 description for <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.
Stores a curve fit table to a file on the specified disk.

**Command Syntax:**

```
MMEMory:STORe:CFIT <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;"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:**

**MMEMory:STORE:DATA:TRACE{1 | 2 | 3 | 4} command**

Saves the specified data table to a file on the specified disk.

**Command Syntax:**

```
MMEMory:STORE:DATA: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;"mem:stor:dat:trac2, 'int:testdtab'"
OUTPUT 711;"MMEM:STORE:DATA:TRAC1, 'MYTABLE'"
```

**Attribute Summary:**

<table>
<thead>
<tr>
<th>Option</th>
<th>not applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Required</td>
<td>no</td>
</tr>
<tr>
<td>Preset State</td>
<td>not applicable</td>
</tr>
<tr>
<td>SCPI Compliance</td>
<td>instrument-specific</td>
</tr>
</tbody>
</table>

**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.

15-28
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>({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: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**

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>{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: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>[{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: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:PROGرام:FORMat**  
command/query

Specifies the format Instrument BASIC programs are stored.

**Command Syntax:**  
MMEMory:STORe:PROGرام:FORMat ASCII|BINary

**Example Statements:**  
OUTPUT 711;"MMEM:STOR:PROG:FORM ASC"  
OUTPUT 711;"mmemory:store:program:format binary"

**Query Syntax:**  
MMEMory:STORe:PROGرام: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 programs 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;"mmemory:store:state 1, '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>
```

```markdown
<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:sysfile.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:

\[
\text{MEMory:STORe:TCAPture} \ <\text{FILENAME}>
\]

\[
\text{FILENAME} ::= '[<\text{PATH}>]<\text{filename}>'
\]

\[
<\text{PATH}> ::= <\text{MSIS}>\{<\text{DOS_DIR}>\}
\]

\[
<\text{MSIS}> ::= \text{RAM:}|\text{NVRAM:}|\text{INT:}
\]

\[
\text{EXT},<\text{select_code}><,\text{unit_number}>]]:
\]

\[
<\text{DOS_DIR}> ::= \text{\textbackslash}<\text{DIR_NAME}>\text{\textbackslash}
\]

\[
<\text{DIR_NAME}> ::= \text{ASCII characters}
\quad\text{(DOS only)}
\]

\[
<\text{filename}> ::= \text{ASCII characters (see beginning of chapter}
\quad\text{for <filename> restrictions)}
\]

Example Statements:

\[
\text{OUTPUT 711;"mmem:stor:tcap, 'int:tcap1'"}
\]

\[
\text{OUTPUT 711;"MMEM:STORE:TCAPTURE, 'MYTIME'"}
\]

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\}, <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: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:STOR:TRAC:FORMat**

**command/query**

Specify the format of data saved to a file.

**Command Syntax:**

```
MMEM:STOR:TRAC:FORM SDF|ASCII
```

**Example Statements:**

```
OUTPUT 711; "MMEM:STOR:TRAC:FORM ASCII"
```

**Query Syntax:**

```
MMEM:STOR:TRAC:FORM?
```

**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>{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: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

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

**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).
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 SELelected mnemonic operate on the active program buffer. Since SELelected 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.
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.
**PROGram:EXPLicit:DEFine**

Loads an Instrument BASIC program into the specified program buffer from an external controller.

**Command Syntax:**

\[
\text{PROGram:EXPLicit:DEFine} \{\text{PROGram1|PROGram2|PROGram3|PROGram4|PROGram5}\}, <\text{PROGRAM}> \\
\]

\[<\text{PROGRAM}> ::= <\text{BLOCK}> \\
<\text{BLOCK}> ::= #<\text{byte}>[<\text{length_bytes}>]<\text{data_bytes}> \\
<\text{byte}> ::= \text{one byte specifying the number of length bytes to follow (ASCII-encoded)} \\
<\text{length_bytes}> ::= \text{bytes specifying the number of data bytes to follow (ASCII-encoded)} \\
<\text{data_bytes}> ::= \text{the bytes that define an Instrument BASIC program} \\
\]

**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:**

\[\text{PROGram:EXPLicit:DEFine?} \]

**Return Format:**

\[\text{definite length } <\text{BLOCK}> \]

**Attribute Summary:**

\[\text{Option: 1C2 Instrument BASIC} \\
\text{Synchronization Required: no} \\
\text{Preset State: not applicable} \\
\text{SCPI Compliance: confirmed} \]

**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 BASIC program 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**

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.
**PROGram[:SELected]:DEFine**

Loads an Instrument BASIC program from an external controller into the active program buffer.

**Command Syntax:**

PROGram[: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:**

PROGram[: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

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.
**PROGram[:SELeted]:DELete[:SELeted] command**

Deletes the active Instrument BASIC program.

**Command Syntax:**

```
PROGram[:SELeted]:DELete[:SELeted]
```

**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:**

\[
\text{PROGram[:SELected]:MALLocate} \{<\text{number}>|<\text{bound}|\text{DEFault}\}
\]

- \(<\text{number}> ::= \text{a real number (NRf data)}\)
  - limits: 1200:500000
- \(<\text{bound}> ::= \text{MAX|MIN}\)

**Example Statements:**

```
OUTPUT 711:"PROG:MALLOCATE 416211"
OUTPUT 711:"PROG:SELECTED:MALL 187046"
```

**Query Syntax:**

\[
\text{PROGram[:SELected]:MALLocate}\?
\]

**Return Format:**

\text{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

**PROGram[:SELeected]:NAME**

**command/query**

Selects an Instrument BASIC program.

**Command Syntax:**

PROGram[:SELeected]:NAME
PROGram1|PROGram2|PROGram3|PROGram4|PROGram5

**Example Statements:**

OUTPUT 711; "prog:seleected:name PROGRAM4"
OUTPUT 711; "Program:Name PROGRAM1"

**Query Syntax:**

PROGram[:SELeecte]: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 (FORM at 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

PROGram[:SELeeted]:STATe command/query

Selects the state of the active Instrument BASIC program.

**Command Syntax:**

```
PROGram[:SELeeted]:STATe STOP|PAUSe|RUN|CONTinue
```

**Example Statements:**

```
OUTPUT 711;""PROG:STATE CONTINUE"
OUTPUT 711;"prog:seleceted:stat RUN"
```

**Query Syntax:**

```
PROGram[:SELeeted]: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.
**PROGrame**:SELelected:STRing command/query

Loads a new value for the specified string variable for the active Instrument BASIC program.

**Command Syntax:**

\[
\text{PROGrame[ :SELelected]} : \text{STRING} \ ' <\text{Variable}> ' , ' <\text{STRING}> ' \\
\]

\(<\text{Variable}> ::= \text{name of string variable} \) \\
(mandatory "$" at the end of the name) \\
\(<\text{STRING}> ::= \text{ASCII characters} \cdot 0 \text{ through 255} \) \\
maximum number of characters: 32766

**Example Statements:**

```
OUTPUT 711; "PROGRAM:SEL:STR 'A$','Done'"
OUTPUT 711; "prog:str 'Message$','Measuring'"
```

**Query Syntax:**

\[
\text{PROGrame[ :SELelected]} : \text{STRING}\
\]

**Return Format:**

" <\text{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.
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:] 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:**

\[ [ \text{SENSe:} ] \text{AVERage:COUNt } <\text{number}>|<\text{step}>|<\text{bound}> \]

- **<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:**

\[ [ \text{SENSe:} ] \text{AVERage:COUNt}? \]

**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:] AVERage:HOLD

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

Enables impulse detection in octave measurements.

Command Syntax:  

\[ \text{[SENSe:]} \text{AVERAGE:IMPulse \ OFF|0|ON|1} \]

Example Statements:  

\[
\begin{align*}
\text{OUTPUT 711;"AVER:IMPULSE ON"} \\
\text{OUTPUT 711;"sens:aver:impulse ON"}
\end{align*}
\]

Query Syntax:  

\[ \text{[SENSe:]} \text{AVERAGE:IMPulse?} \]

Return Format:  

\text{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:] AVERAGE:IRES:RESULT:RATE command/query

Specifies how often the display is updated when fast average mode is on.

Command Syntax:  
[SENSe:]AVERAGE:IRES:RESULT: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:IRES:RESULT: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: \[ S\ E\ N\ S\ e\ : ]AVERage:IRESult[:STATe] OFF|0|ON|1

Example Statements: OUTPUT 711;":aver:i res ON"
OUTPUT 711;"Sense:Aver:i result:Stat OFF"

Query Syntax: \[ S\ E\ N\ S\ e\ : ]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:]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.
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:] AVERage:PREView:REJect  

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:] 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.
Figure 18-1. Transition of Operation Status Register

Bits When AVERAGE:STATE] ON.
[SENSe:]AVERage:TCONtrol

Specifies how the analyzer behaves after the count (AVER:COUN) is reached.

**Command Syntax:**

```
[SENSe:]AVERAGE:TCON trol FREEze|REPeat|EXPonential
```

**Example Statements:**

```
OUTPUT 711; "Sense:Aver:Tcon FREEZE"
OUTPUT 711; "AVERAGE:TCON REPEAT"
```

**Query Syntax:**

```
[SENSe:]AVERAGE:TCON trol?
```

**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**

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>valid</td>
</tr>
<tr>
<td>RMS</td>
<td>valid</td>
</tr>
<tr>
<td>TIME</td>
<td>valid</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>

**Correlation instrument mode**

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>not allowed</td>
</tr>
<tr>
<td>RMS</td>
<td>valid</td>
</tr>
<tr>
<td>TIME</td>
<td>valid</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>

**Histogram instrument mode**

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>not allowed</td>
</tr>
<tr>
<td>RMS</td>
<td>not allowed</td>
</tr>
<tr>
<td>TIME</td>
<td>valid</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>

**Octave instrument mode**

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>valid</td>
</tr>
<tr>
<td>RMS</td>
<td>valid</td>
</tr>
<tr>
<td>TIME</td>
<td>not allowed</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>
### Order instrument mode

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>not allowed</td>
</tr>
<tr>
<td>RMS</td>
<td>not allowed</td>
</tr>
<tr>
<td>TIME (time)</td>
<td>valid</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>

### Swept Sine instrument mode

<table>
<thead>
<tr>
<th>AVERage:TYPE</th>
<th>AVERage:TCONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FREeze</td>
</tr>
<tr>
<td>MAX</td>
<td>not allowed</td>
</tr>
<tr>
<td>RMS</td>
<td>not allowed</td>
</tr>
<tr>
<td>TIME</td>
<td>not allowed</td>
</tr>
<tr>
<td>ECONfidence</td>
<td>not allowed</td>
</tr>
</tbody>
</table>
**SENSe: AVERage:TIME**  
**command/query**

Specify the time period used in averaging octave measurements and histograms.

**Command Syntax:**

```
[ SENSep: ] 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:**

```
[ SENSep: ] 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.
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 bins2 (specified with the HIST:BINS command).

$$\text{Optimal Histogram} = (\text{AVER:TIME}) = (\text{HIST: BINS})^2$$
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.
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**

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:]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 / ((Stop frequency) − (Start frequency)).
[SENSe:]DATA:HEADer:FREQuency:STOP?

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:]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

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:**

Any data values larger than the range value will be clipped.
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:**  
\[
\text{[SENSe:]FREQuency:BLOCksize <number>|<step>|<bound>}
\]

- **<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:**  
\[
\text{[SENSe:]FREQuency:BLOCksize?}
\]

**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.
### Table 18-2. Number of displayed data points according to blocksize

#### FFT Instrument Mode

<table>
<thead>
<tr>
<th>Blocksize</th>
<th>Baseband Number of real time points (time data)</th>
<th>Zoom (Start frequency = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of frequency points (frequency data)</td>
<td>Number of frequency points (frequency data)</td>
</tr>
<tr>
<td>256</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>512</td>
<td>201</td>
<td>201</td>
</tr>
<tr>
<td>1024</td>
<td>401</td>
<td>401</td>
</tr>
<tr>
<td>2048</td>
<td>801</td>
<td>801</td>
</tr>
</tbody>
</table>

#### Correlation Instrument Mode

(no complex data)

<table>
<thead>
<tr>
<th>Blocksize</th>
<th>Time domain (real data)</th>
<th>Auto- and cross correlation (real data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0 to T/2</td>
</tr>
<tr>
<td>256</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>512</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>1024</td>
<td></td>
<td>512</td>
</tr>
<tr>
<td>2048</td>
<td></td>
<td>1024</td>
</tr>
</tbody>
</table>
SENSe:]FREQuency:CENTer

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:cent 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:]FREQuency:MANual

Selects a discrete point to be measured during manual sweep mode.

Command Syntax:  

\[[\text{SENSe:}]\text{FREQuency:MANual} \{\text{<number>}[\text{<unit>}]|\text{<step>}|\text{<bound>}\]\n
\(<\text{number}> ::= \text{a real number} \text{ (NRf data)} \text{ limits: 0.015625:51200.0}\n\)<\text{unit}> ::= [\text{HZ}]\n\)<\text{step}> ::= \text{UP|DOWN}\n\)<\text{bound}> ::= \text{MAX|M IN}\n
Example Statements:  

\text{OUTPUT 711;"FREQUENCY:MAN 1388.13"} 
\text{OUTPUT 711;"sense:freq:man 24158.9"}\n
Query Syntax:  

[\text{SENSe:}]\text{FREQuency:MANual?}\n
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:]FREQuency:RESolution command/query

Specifies the frequency measurement resolution for FFT and swept sine instrument modes.

Command Syntax:

```
[SENSe:]FREQuency:RESolution {<number>[<unit>]}|<step>|
<bound>
```

- `<number>` ::= a real number (NRf data)
  limits: 0.015625:51200.0
- `<unit>` ::= [HZ|PCT|PNT/SWP|PNT/DEC|PNT/OCT]
  (swept sine only)
- `<step>` ::= UP|DOWN
- `<bound>` ::= MAX|MIN

Example Statements:

```
OUTPUT 711;"SENS:FREQ:RES 100 PNT/SWP"
OUTPUT 711;":freq:res 800"
```

Query Syntax:

```
[SENSe:]FREQuency:RESolution?
```

Return Format:

Real

Attribute Summary:

- Option: not applicable
- Synchronization Required: no
- Preset State: 400 (FFT and correlation)
  101 PNT/SWP (swept sine)
- SCPI Compliance: confirmed

Description:

In FFT analysis instrument mode (INST:SEL FFT):

Variable resolution is available and can be set to 100, 200, 400 or 800 lines. Frequency resolution is unitless. See table 18-3 to determine the number of displayed data points with variable resolution. This command is similar to FREQuency:BLOCksize which specifies the frequency resolution by setting the length of the time record.
Table 18.3. Number of displayed data points in variable resolution

### FFT Instrument Mode

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Baseband</th>
<th>Zoom (Start frequency ( \sqrt{0} ))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of frequency points (frequency data)</td>
<td>Number of real time points (time data)</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>256</td>
</tr>
<tr>
<td>200</td>
<td>201</td>
<td>512</td>
</tr>
<tr>
<td>400</td>
<td>401</td>
<td>1024</td>
</tr>
<tr>
<td>800</td>
<td>801</td>
<td>2048</td>
</tr>
</tbody>
</table>

### Swept Sine Instrument Mode

(FREQ:RES:AUTO ON)

<table>
<thead>
<tr>
<th>FREQ:RES:AUTO:MIN (Number of measured points (^1))</th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

\(^1\) 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

<table>
<thead>
<tr>
<th>FREQ:RES (Number of measured points)</th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

1. 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.
**[SENSe:]**

**FREQ:RES:AUTO**

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)

<table>
<thead>
<tr>
<th>FREQ:RES:AUTO:MIN (Number of measured points)</th>
<th>Number of displayed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 401 PNT/SWP</td>
<td>401</td>
</tr>
<tr>
<td>402 to 801 PNT/SWP</td>
<td>801</td>
</tr>
</tbody>
</table>

1 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:]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.
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.
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)

<table>
<thead>
<tr>
<th></th>
<th>1 Channel (INPut2 OFF)</th>
<th>2 Channel (INPut2 ON;INPut4 OFF)</th>
<th>4 Channel (INPut4 ON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1 Octave</td>
<td>63 mHz to 16 kHz</td>
<td>63 mHz to 8 kHz</td>
<td>63 mHz - 4 kHz</td>
</tr>
<tr>
<td>1/3 Octave</td>
<td>100 mHz to 40 kHz</td>
<td>100 mHz to 20 kHz</td>
<td>100 mHz to 10 kHz</td>
</tr>
<tr>
<td>1/12 Octave</td>
<td>99.7 mHz to 12.3 kHz</td>
<td>99.7 mHz to 6.17 kHz</td>
<td>99.7 mHz to 3.08 kHz</td>
</tr>
</tbody>
</table>
[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)
- `<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:

\[(\text{maximum frequency span}) / 2^n\]

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:

\[\text{SWE:TIME} = \frac{\text{FREQ:RES}}{\text{FREQ:SPAN}}\]

The frequency span limits the range of the start frequency values (FREQ:STAR) according to the following formulas:

- for one channel measurements:
  \[(\text{FREQ:STAR}) \leq 115 \text{ kHz} - \frac{\text{FREQ:SPAN}}{2}\]
- for two channel measurements:
  \[(\text{FREQ:STAR}) \leq 57.5 \text{ kHz} - \frac{\text{FREQ:SPAN}}{2}\]
- for four channel measurements:
  \[(\text{FREQ:STAR}) \leq 28.75 \text{ kHz} - \frac{\text{FREQ:SPAN}}{2}\]
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.

\[ \text{FREQ:SPAN} = \frac{\text{FREQ:RES}}{\text{SWE:TIME}} \]

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.

\[ \text{frequency resolution} = \frac{\text{FREQ:SPAN}}{400} \]

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:

\[ (\text{FREQ:STAR}) \geq 51.2 \text{ kHz} - (\text{FREQ:SPAN}) \]

for four channel measurements:

\[ (\text{FREQ:STAR}) \geq 25.6 \text{ kHz} - (\text{FREQ:SPAN}) \]

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.
Table 18-7. Range of Available Frequency Spans in FFT analysis instrument mode (in hertz)

<table>
<thead>
<tr>
<th>Type of Analyzer</th>
<th>Type of Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Channel (INPut2 OFF)</td>
</tr>
<tr>
<td>Standard 2 Channel</td>
<td>195 mHz - 102.4 kHz</td>
</tr>
<tr>
<td>Option AYG 4 Channel</td>
<td>AAF off (INPut:FILTer OFF)</td>
</tr>
<tr>
<td></td>
<td>195 mHz - 102.4 kHz</td>
</tr>
</tbody>
</table>

* 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):**

<table>
<thead>
<tr>
<th>Number of Channels</th>
<th>Start Frequency</th>
<th>Stop Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 channel (without option AY6)</td>
<td>0 Hz</td>
<td>102.4 kHz</td>
</tr>
<tr>
<td>2 channel (without option AY6)</td>
<td>0 Hz</td>
<td>51.2 kHz</td>
</tr>
<tr>
<td>1 channel (with option AY6)</td>
<td>0 Hz</td>
<td>51.2 kHz</td>
</tr>
<tr>
<td>2 channel (with option AY6)</td>
<td>0 Hz</td>
<td>51.2 kHz</td>
</tr>
<tr>
<td>4 channel (with option AY6)</td>
<td>0 Hz</td>
<td>25.6 kHz</td>
</tr>
</tbody>
</table>

**In Octave analysis instrument mode (INST:SEL OCT; Option 1D1):**

<table>
<thead>
<tr>
<th>Number of Channels</th>
<th>Start Frequency</th>
<th>Stop Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 channel - 1/1</td>
<td>16 Hz</td>
<td>16 kHz</td>
</tr>
<tr>
<td>1 channel - 1/3</td>
<td>12.5 Hz</td>
<td>40 kHz</td>
</tr>
<tr>
<td>1 channel - 1/12</td>
<td>6.3825 Hz</td>
<td>12.338 kHz</td>
</tr>
<tr>
<td>2 channel - 1/1</td>
<td>16 Hz</td>
<td>8 kHz</td>
</tr>
<tr>
<td>2 channel - 1/3</td>
<td>12.5 Hz</td>
<td>20 kHz</td>
</tr>
<tr>
<td>2 channel - 1/12</td>
<td>6.3825 Hz</td>
<td>6.169 kHz</td>
</tr>
<tr>
<td>4 channel (Option AY6) - 1/1</td>
<td>16 Hz</td>
<td>4 kHz</td>
</tr>
<tr>
<td>4 channel (Option AY6) - 1/3</td>
<td>12.5 Hz</td>
<td>10 kHz</td>
</tr>
<tr>
<td>4 channel (Option AY6) - 1/12</td>
<td>6.3825 Hz</td>
<td>3.084 kHz</td>
</tr>
</tbody>
</table>
In **Correlation analysis instrument mode** *(INST:SEL CORR)*:

<table>
<thead>
<tr>
<th>Number of Channels</th>
<th>Record Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 channel</td>
<td>7.8125 mS</td>
</tr>
<tr>
<td>2 channel</td>
<td>7.8125 mS</td>
</tr>
<tr>
<td>4 channel</td>
<td>15.625 mS</td>
</tr>
</tbody>
</table>

In **Histogram/Time instrument mode** *(INST:SEL HIST)*:

<table>
<thead>
<tr>
<th>Number of Channels</th>
<th>Record Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 channel (without option AY6)</td>
<td>3.9062 mS</td>
</tr>
<tr>
<td>2 channel (without option AY6)</td>
<td>7.8125 mS</td>
</tr>
<tr>
<td>1 channel (with option AY6)</td>
<td>7.8125 mS</td>
</tr>
<tr>
<td>2 channel (with option AY6)</td>
<td>7.8125 mS</td>
</tr>
<tr>
<td>4 channel (with option AY6)</td>
<td>15.625 mS</td>
</tr>
</tbody>
</table>
[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:** STAR
- **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:]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):
  \[ 0 \leq \text{start frequency} \leq (115 \text{ kHz} \cdot (\text{frequency span}/2)) \]

- two channel measurements or single channel measurements with option AY6:
  \[ 0 \leq \text{start frequency} \leq (57.5 \text{ kHz} \cdot (\text{frequency span}/2)) \]

- four channel measurements:
  \[ 0 \leq \text{start frequency} \leq (28.75 \text{ kHz} \cdot (\text{frequency span}/2)) \]

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.
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.
Specifies the step size which is used when changing frequency parameters.

**Command Syntax:**

\[
\text{[SENSe:]}\text{FREQuency:STEP[:INCRement]} \{<\text{number}>[<\text{unit}>]|<\text{step}>|<\text{bound}>\}
\]

- \(<\text{number}>\) ::= a real number (NRf data)
  - limits: 0.015625:102400.0
- \(<\text{unit}>\) ::= [HZ|CPM]
- \(<\text{step}>\) ::= UP|DOWN
- \(<\text{bound}>\) ::= MAX|MIN

**Example Statements:**

- OUTPUT 711;":FREQ:STEP 5e3"
- OUTPUT 711;"sens:freq:step:incr 100"

**Query Syntax:**

\[
\text{[SENSe:]}\text{FREQuency:STEP[:INCRement]}?\]

**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

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:]HISTogram:BIN

Command/Query

Specifies the number of bins in a histogram.

Command Syntax:

\[ [\text{SENSe:}] \text{HISTogram:BIN} <\text{number}> | <\text{step}> | <\text{bound}> \]

\(<\text{number}> ::= \text{a real number } (\text{NRf data}) \]
\(<\text{limits: } 4:1024> \]
\(<\text{step}> ::= \text{UP|DOWN} \]
\(<\text{bound}> ::= \text{MAX|MIN} \]

Example Statements:

\[ \text{OUTPUT 711;"HIST:BIN 10"} \]
\[ \text{OUTPUT 711;"SENSE:HISTOGRAM:BIN 400"} \]

Query Syntax:

\[ [\text{SENSe:}] \text{HISTogram:BIN?} \]

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.

\[
\text{Optimal Histogram} = \left( \text{HIST:BIN} \right) = \sqrt{[\text{SENSe:}]\text{AVERage:TIME}}
\]
SENSe:]ORDer:MAXimum

Specifies the number of orders to be displayed.

Command Syntax:  
\[ \text{SENSe:} \text{ORDer:MAXi mum \{<number>[\ <unit>]\}|<step>|<bound>} \]

\(<\text{number}>\ ::= \text{a real number (NRf data)}\)
\(<\text{unit}>\ ::= [\text{ORD}]\)
\(<\text{step}>\ ::= \text{UP|DOWN}\)
\(<\text{bound}>\ ::= \text{MAX|MIN}\)

Example Statements:
\begin{itemize}
  \item OUTPUT 711; "sens:ord:maximum 23.7537"
  \item OUTPUT 711; "Ord:Maximum 147.43"
\end{itemize}

Query Syntax:  
\[ \text{SENSe:} \text{ORDer:MAXi mum?} \]

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:

\[ \text{ORD: MAX \ @ \ 200 (ORD:RES) } \]
[SENSe:] ORDeR:RESoLution

Specifies order resolution.

**Command Syntax:**

```
[SENSe:] ORDeR:RESoLution {<number>[<unit>] | <step> | <bound>}
```

- `<number>` ::= a real number (NRf data)
- `<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.

```
ORD: MAX \leq 200 (ORD:RES)
```

Values can range from .0078125 to 1.
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:] 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; "ORDER: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 (ORDER:RPM:MIN).

If the value you specify for ORDER:RPM:MAX is less than the value set for ORDER:RPM:MIN, the value for ORDER:RPM:MIN is set to equal the value of ORDER:RPM:MAX.

If

```
ORDER:RPM:MAX < ORDER:RPM:MIN
```

then

```
ORDER:RPM:MIN = ORDER:RPM:MAX
```

If the value you specify for ORDER:RPM:MIN is greater than the value set for ORDER:RPM:MAX, the value for ORDER:RPM:MAX is set to equal the value of ORDER:RPM:MIN.

If

```
ORDER:RPM:MIN > ORDER:RPM:MAX
```

then

```
ORDER:RPM:MAX = ORDER:RPM:MAX
```
**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
```
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:ord: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:]REFerence

Specifies the reference channel(s).

Command Syntax:  

\[ \text{SENSe:} \] \text{REFerence SING|e| PAIR} 

Example Statements:  

\begin{align*} 
\text{OUTPUT 711;} & \text{ "Sense:Ref PAIR"} \\
\text{OUTPUT 711;} & \text{ "REFERENCE PAIR"} 
\end{align*} 

Query Syntax:  

\[ \text{SENSe:} \] \text{REFerence?} 

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:Stat 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:] SWEep:DIRection

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:DWELI

Specifies the integration time for swept sine measurements.

Command Syntax:

\[
[\text{SENSe:}] \text{SWEEP } \langle \text{dwell\_time} \rangle
\]

\[
\langle \text{dwell\_time} \rangle \ ::= \ {\langle \text{number}\rangle[\langle \text{unit}\rangle]}|\langle \text{step}\rangle|\langle \text{bound}\rangle
\]

\[
\langle \text{number}\rangle \ ::= \ \text{a real number (NRf data)}
\]

- limits: 250e-6:32768 S
- or
- 1:234 CYCLE

\[
\langle \text{unit}\rangle \ ::= \ \{\text{S}|\text{CYCLE}\}
\]

\[
\langle \text{step}\rangle \ ::= \ \text{UP}|\text{DOWN}
\]

\[
\langle \text{bound}\rangle \ ::= \ \text{MAX}|\text{MIN}
\]

Example Statements:

- OUTPUT 711; "sense:sweep:dwel .005 s"
- OUTPUT 711; "Swe:Dwell 1 cycle"

Query Syntax:

\[
[\text{SENSe:}] \text{SWEep:DWELI}\ ?
\]

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:]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:]SWEp:OVERlap

Specifies the maximum amount of time record overlap.

**Command Syntax:**

```plaintext
[SENSe:]SWEp:OVERlap {<number>[<unit>]}|<step>|<bound>
```

- `<number>` := a real number (NRf data)
- `<unit>` := [PCT]
- `<step>` := UP|DOWN
- `<bound>` := MAX|MIN

**Example Statements:**

```
OUTPUT 711;"sens:swe:overlap 52"
OUTPUT 711;"Swe:Overlap 26"
```

**Query Syntax:**

```plaintext
[SENSe:]SWEp: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.
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 an 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 LINER"
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:]SWEep:STI Me command/query

Specifies the settling time for a swept sine measurement.

Command Syntax:

```
[S E N S e: ] S W E e p: S T I Me \{ <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:SWEP:STIM 0.005 s"
OUTPUT 711; "SWEP:STIM 1 cycle"
```

Query Syntax:

```
[S E N S e: ] S W E e p: S T I Me ?
```

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).
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:]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:]TCAPture:FILE

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:]TCAPture:LENGth  

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;"T:capture: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:

<table>
<thead>
<tr>
<th>Maximum Capture Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>(in blocks of 1024 sample points)</td>
</tr>
<tr>
<td>1 Channel</td>
</tr>
<tr>
<td>Standard</td>
</tr>
<tr>
<td>Add 4 Mbytes RAM (Option AN2)</td>
</tr>
<tr>
<td>Add 8 Mbytes RAM (Option UFC)</td>
</tr>
</tbody>
</table>
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:] TCAPture:MALlocate command

Allocates memory for the time capture buffer.

Command Syntax: [SENSe:] TCAPture: MALlocate

Example Statements:
OUTPUT 711; "sense:tcp:mall"
OUTPUT 711; "Tcapture:Mal1"

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

---

18-75
[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

Specifies the tachometer’s maximum RPM when included in the time capture buffer.

Command Syntax:

\[
\text{[SENSe:]} \text{TCAPture:TACHometer:RPM:MAXimum <number> | <step> | <bound>}
\]

\[
\text{<number> ::= a real number (NRf data)}
\]

\[
\text{limits: 5:491519}
\]

\[
\text{<step> ::= UP|DOWN}
\]

\[
\text{<bound> ::= MAX|MIN}
\]

Example Statements:

\[
\begin{align*}
\text{OUTPUT 711;"TCAPTURE:TACH:RPM:MAXIMUM 193783"} \\
\text{OUTPUT 711;"sens:tcapture:tach:rpm:maximum 275254"}
\end{align*}
\]

Query Syntax:

\[
\text{[SENSe:]TCAPture:TACHometer:RPM:MAXimum?}
\]

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:]TCAPture:TACHometer[:STATe]

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.
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.
Assigns a name to the transducer units for the specified input channel when VOLT:RANG:UNIT:XDCR:LABel is USER.

**Command Syntax:**

```
[SENSe:]VOLTage[1|2|3|4][:DC]: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.
Specifies the transducer sensitivity for transducer units.

**Command Syntax:**

\[
[\text{SENSe:}] \text{VOLTage}\{1|2|3|4\}:\text{DC}:\text{RANGE}:\text{UNIT}:\text{USER}:\text{SFACtor} \\
\{\text{<number>}[\text{<unit>}]\}|\text{<step>}|\text{<bound>}
\]

- **<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; "SENSE: VOLTAGE3: DC: RANGE: UNIT: USER: SFACtor 1.8e-06"

**Query Syntax:**

\[
[\text{SENSe:}] \text{VOLTage}\{1|2|3|4\}:\text{DC}:\text{RANGE}:\text{UNIT}:\text{USER}:\text{SFACtor}\ ?
\]

**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.
Enables the use of transducer units.

**Command Syntax:**

```
[SENSe:]VOLTage[1|2|3|4][:DC]:RANGe:UNIT:USER[:STATe]
```

**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.
   - 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 second², kilograms, Newtons, dynes, inches, inches per second, inches per second², pounds, and mils.
   - 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.


3. Enable the transducer unit setup with VOLT:RANG:UNIT:USER ON.
Specifies a transducer unit for the specified channel.

**Command Syntax:**

[SENSe:]VOLTage[1|2|3|4]:RANGE:UNIT:XDCR:LABel

PA|G|M/S2|M/S|M|KG|N|DY|N|INCH/S2|INCH/S|INCH|MIL|LB|USER

**Example Statements:**

OUTPUT 711; "SENSE: VOLT2: RANG: UNIT: XDCR: LABEL PA"

**Query Syntax:**

[SENSe:]VOLTage[1|2|3|4]:RANGE:UNIT:XDCR:LABel?

**Return Format:**

PA|G|M/S2|M/S|M|KG|N|DY|N|INCH/S2|INCH/S|INCH|MIL|LB|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.

<table>
<thead>
<tr>
<th>Unit of Measure</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>microphones</td>
<td>Pascals</td>
</tr>
<tr>
<td>acceleration</td>
<td>g's</td>
</tr>
<tr>
<td>displacement</td>
<td>mils</td>
</tr>
<tr>
<td>impact hammer</td>
<td>pound force</td>
</tr>
<tr>
<td>shakers</td>
<td>Newtons</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit of Measure</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VOLT: RANG: UNIT: XDCR PA</td>
</tr>
<tr>
<td></td>
<td>VOLT: RANG: UNIT: XDCR G</td>
</tr>
<tr>
<td></td>
<td>VOLT: RANG: XDCR MIL</td>
</tr>
<tr>
<td></td>
<td>VOLT: RANG: XDCR LB</td>
</tr>
<tr>
<td></td>
<td>VOLT: RANG: XDCR N</td>
</tr>
</tbody>
</table>

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:]VOLTage[1 | 2 | 3 | 4][:DC]:RANGe:UPPer

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:

$$e^{-t/t}$$

Where

- $t$ is the position (in time) in the record length
- $t$ 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 $t=0$. 

18-87
[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;":WIND: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.
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:]WINDow[1|2|3|4]:TYPE

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**

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

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|PI NK|USER|CAPT

Example Statements:
OUTPUT 711;":Source:Func CAPT"
OUTPUT 711;"SOUR:FUNCTION:SHAPE 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

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.
SOUR:USER:REPeat command/query

Enables the source repeat function.

**Command Syntax:**
SOUR:USER:REPeat OFF|0|ON|1

**Example Statements:**
OUTPUT 711;"Sour:User:Repeat OFF"
OUTPUT 711;"SOUR:USER:REP ON"

**Query Syntax:**
SOUR: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).$

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
- SOUR: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:VOLT: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][:AMPplitude] command) and the DC offset exceed ±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).
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:VOLT:LEVEL: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: \(-9.9\times10^{-37}:13.9794\)
- `<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

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;"S: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  

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
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.

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.
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.)

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.

STATUs

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.

STATus:DEVice:NTRansition command/query

Sets and queries bits in the Device Status negative transition register.

**Command Syntax:**

```plaintext
STATus:DEVice:NTRansition <number>|<bound>
```

- `<number>` ::= a real number (NRf data)
  - limits: 0:32767
- `<bound>` ::= MAX|MIN

**Example Statements:**

```plaintext
OUTPUT 711;":stat:dev:ntr 4"
OUTPUT 711;"STATUS:DEVICE:NTR 17"
```

**Query Syntax:**

```plaintext
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.

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.

STATus:OPERation:CONDition?

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.)

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.

STATus:OPERation[:EVENt]?

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.

STATus

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

Sets bits in the Operation Status positive transition register.

Command Syntax:

\[
\text{STATus:OPERation:PTRansition } \langle\text{number}\rangle|\langle\text{bound}\rangle
\]

\(<\text{number}\> ::= \text{a real number (NRf data)}
\]

\(<\text{bound}\> ::= \text{MAX | MIN}
\]

Example Statements:

\begin{itemize}
\item OUTPUT 711; "STAT:OPER:PTR 1536"
\item OUTPUT 711; "STAT:OPER:PTR 2048"
\end{itemize}

Query Syntax:

\[
\text{STATus:OPERation:PTRansition?}
\]

Return Format:

\[
\text{Integer}
\]

Attribute Summary:

\begin{itemize}
\item Option: not applicable
\item Synchronization Required: no
\item Preset State: not affected by Preset
\item SCPI Compliance: confirmed
\end{itemize}

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.

STATus

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?

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

**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:QUES:tionable[:EVEN]? query

Reads and clears the Questionable Status event register.

**Query Syntax:**

```plaintext
STATus:QUES:tionable[:EVEN]? 
```

**Example Statements:**

- `OUTPUT 711;":status:ques?"`
- `OUTPUT 711;"Sta: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?

Reads and clears the Limit Fail condition register.

Query Syntax:

```
STATus:QUEStionable:LI Mit:CONDition?
```

Example Statements:

```
OUTPUT 711; "STATUS:QUE:LI:COND?"
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.)

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 *GPIB 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:Limi?"
OUTPUT 711;"STAT:QUES:LIMIT:EVEN?"

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.

STATus:QUEStionable:LIMit:NTRansition command/query

Sets and queries bits in the Limit Fail negative transition register.

**Command Syntax:**

```plaintext
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:**

```plaintext
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.

STATus

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.

**STATus:QUEStionable:NTRansition**

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:ntr 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.
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?

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|M IN

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:ntr 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:**

\[ \text{STATus:QUESTionable:VOLTage:PTRansition} \ <\text{number}>|<\text{bound}> \]

\(<\text{number}> ::= \text{a real number (NRf data)} \]
\[ \text{limits: } 0:32767 \]
\n\(<\text{bound}> ::= \text{MAX|MIN} \]

**Example Statements:**

OUTPUT 711;":STAT:QUES:VOLTAGE:PTR 2"
OUTPUT 711;"status:ques:volt:ptr 3"

**Query Syntax:**

\[ \text{STATus:QUESTionable:VOLTage:PTRansition}? \]

**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:**

```plaintext
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:**

```plaintext
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.

STATus:USER[:EVENt]?

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.

STATus:USER:PULSe

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:脉se 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.
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**

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

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.
**SYST:COMMunicate:SERial[:RECeive]:BAUD**

Specifies the data-transfer rate between the analyzer and RS-232-C peripheral devices.

**Command Syntax:**

```
SYST: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:**

```
SYST: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:**
```plaintext
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:**
```plaintext
SYSTem: COMMunicate: SERial [: REceive]: PACE?
```

**Return Format:**
```plaintext
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.
SYSTM: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 \textit{command/query}

Sets the RS-232 transmitter handshake pacing type.

\textbf{Command Syntax:} \texttt{SYSTem: COMMunicate: SERial: TRANsmit: PACE NONE|XON|DSR}

\textbf{Example Statements:} \texttt{OUTPUT 711; "SYST: COMM: SER: TRAN: PACE XON"}
\texttt{OUTPUT 711; "system: comm: serial: transmit: pace none"}

\textbf{Query Syntax:} \texttt{SYSTem: COMMunicate: SERial: TRANsmit: PACE?}

\textbf{Return Format:} NONE|XON|DSR

\textbf{Attribute Summary:}
- Option: not applicable
- Synchronization Required: no
- Preset State: not affected by Preset
- SCPI Compliance: confirmed, except DSR

\textbf{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:**

```plaintext
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:**

```plaintext
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]**

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Keycode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>29</td>
</tr>
<tr>
<td>Disp Format</td>
<td>39</td>
</tr>
<tr>
<td>Analys</td>
<td>18</td>
</tr>
<tr>
<td>Trace Coord</td>
<td>23</td>
</tr>
<tr>
<td>Active Trace</td>
<td>4</td>
</tr>
<tr>
<td>Meas Data</td>
<td>2</td>
</tr>
<tr>
<td>Window</td>
<td>30</td>
</tr>
<tr>
<td>Freq</td>
<td>17</td>
</tr>
<tr>
<td>Inst Mode</td>
<td>1</td>
</tr>
<tr>
<td>Trigger</td>
<td>32</td>
</tr>
<tr>
<td>Source</td>
<td>22</td>
</tr>
<tr>
<td>Input</td>
<td>3</td>
</tr>
<tr>
<td>Avg</td>
<td>55</td>
</tr>
<tr>
<td>Pause/Cont</td>
<td>19</td>
</tr>
<tr>
<td>Start</td>
<td>6</td>
</tr>
<tr>
<td>System Utility</td>
<td>24</td>
</tr>
<tr>
<td>Disk Utility</td>
<td>33</td>
</tr>
<tr>
<td>Save/Recall</td>
<td>13</td>
</tr>
<tr>
<td>Help</td>
<td>54</td>
</tr>
<tr>
<td>Preset</td>
<td>31</td>
</tr>
<tr>
<td>Local/GPIB</td>
<td>45</td>
</tr>
<tr>
<td>Plot/Print</td>
<td>16</td>
</tr>
<tr>
<td>BASIC</td>
<td>35</td>
</tr>
<tr>
<td>F1</td>
<td>37</td>
</tr>
<tr>
<td>F2</td>
<td>40</td>
</tr>
<tr>
<td>F3</td>
<td>52</td>
</tr>
<tr>
<td>F4</td>
<td>14</td>
</tr>
<tr>
<td>F5</td>
<td>36</td>
</tr>
<tr>
<td>F6</td>
<td>21</td>
</tr>
<tr>
<td>F7</td>
<td>20</td>
</tr>
<tr>
<td>F8</td>
<td>53</td>
</tr>
<tr>
<td>F9</td>
<td>43</td>
</tr>
<tr>
<td>Rtn</td>
<td>38</td>
</tr>
</tbody>
</table>
## Front Panel Keycodes

**by Front Panel Keys**

<table>
<thead>
<tr>
<th>Key</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>1</td>
<td>51</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>48</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>46</td>
</tr>
<tr>
<td>. (decimal point)</td>
<td>9</td>
</tr>
<tr>
<td>+/-</td>
<td>42</td>
</tr>
<tr>
<td>Back Space</td>
<td>25</td>
</tr>
<tr>
<td>Marker Value</td>
<td>34</td>
</tr>
<tr>
<td>i</td>
<td>28</td>
</tr>
<tr>
<td>Å</td>
<td>26</td>
</tr>
<tr>
<td>Marker</td>
<td>50</td>
</tr>
<tr>
<td>Marker Fctn</td>
<td>10</td>
</tr>
</tbody>
</table>
Front Panel Keycodes
by Keycode

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Inst Mode</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Meas Data</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Input</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Active Trace</td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>Start</td>
</tr>
<tr>
<td><strong>9</strong></td>
<td>. (decimal point)</td>
</tr>
<tr>
<td><strong>10</strong></td>
<td>Marker Fctn</td>
</tr>
<tr>
<td><strong>11</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>12</strong></td>
<td>8</td>
</tr>
<tr>
<td><strong>13</strong></td>
<td>Save/Recall</td>
</tr>
<tr>
<td><strong>14</strong></td>
<td>F4</td>
</tr>
<tr>
<td><strong>15</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>16</strong></td>
<td>Plot/Print</td>
</tr>
<tr>
<td><strong>17</strong></td>
<td>Freq</td>
</tr>
<tr>
<td><strong>18</strong></td>
<td>Analys</td>
</tr>
<tr>
<td><strong>19</strong></td>
<td>Pause/Cont</td>
</tr>
<tr>
<td><strong>20</strong></td>
<td>F7</td>
</tr>
<tr>
<td><strong>21</strong></td>
<td>F6</td>
</tr>
<tr>
<td><strong>22</strong></td>
<td>Source</td>
</tr>
<tr>
<td><strong>23</strong></td>
<td>Trace Coord</td>
</tr>
<tr>
<td><strong>24</strong></td>
<td>System Utility</td>
</tr>
<tr>
<td><strong>25</strong></td>
<td>Back Space</td>
</tr>
<tr>
<td><strong>26</strong></td>
<td>⊕</td>
</tr>
<tr>
<td><strong>28</strong></td>
<td>[</td>
</tr>
<tr>
<td><strong>29</strong></td>
<td>Scale</td>
</tr>
<tr>
<td><strong>30</strong></td>
<td>Window</td>
</tr>
<tr>
<td><strong>31</strong></td>
<td>Preset</td>
</tr>
<tr>
<td><strong>32</strong></td>
<td>Trigger</td>
</tr>
<tr>
<td><strong>33</strong></td>
<td>Disk Utility</td>
</tr>
<tr>
<td><strong>34</strong></td>
<td>Marker Value</td>
</tr>
<tr>
<td><strong>35</strong></td>
<td>BASIC</td>
</tr>
<tr>
<td><strong>36</strong></td>
<td>F5</td>
</tr>
<tr>
<td><strong>37</strong></td>
<td>F1</td>
</tr>
<tr>
<td><strong>38</strong></td>
<td>Rtn</td>
</tr>
<tr>
<td><strong>39</strong></td>
<td>Disp Format</td>
</tr>
<tr>
<td><strong>40</strong></td>
<td>F2</td>
</tr>
<tr>
<td><strong>41</strong></td>
<td>6</td>
</tr>
<tr>
<td><strong>42</strong></td>
<td>+/-</td>
</tr>
<tr>
<td><strong>43</strong></td>
<td>F9</td>
</tr>
<tr>
<td><strong>44</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>45</strong></td>
<td>Local/GPIB</td>
</tr>
</tbody>
</table>
### Front Panel Keycodes

by Keycode

<table>
<thead>
<tr>
<th>Keycode</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>9</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>7</td>
</tr>
<tr>
<td>49</td>
<td>4</td>
</tr>
<tr>
<td>50</td>
<td>Marker</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
</tr>
<tr>
<td>52</td>
<td>F3</td>
</tr>
<tr>
<td>53</td>
<td>F8</td>
</tr>
<tr>
<td>54</td>
<td>Help</td>
</tr>
<tr>
<td>55</td>
<td>Avg</td>
</tr>
</tbody>
</table>
SYSTem:KLOCk

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.
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

SYSTem:PRESet command

Returns most of the analyzer’s parameters to their preset states.

Command Syntax: SYSTem:PRESet

Example Statements:
OUTPUT 711; "Syst:Pres"
OUTPUT 711; "SYSTEM:PRE"n

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:ADD)
- 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

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:

\[ \text{SYSTeM:TIME} \ <\text{hour}>, <\text{minute}>, <\text{second}> \]

\[ <\text{hour}> ::= [ <\text{number}| <\text{bound}> ] \]
\[ <\text{number}> ::= \text{a real number (NRf data)} \]
\[ \text{limits: 0:23} \]
\[ <\text{bound}> ::= \text{MAX|MIN} \]
\[ <\text{minute}> ::= [ <\text{number}| <\text{bound}> ] \]
\[ <\text{number}> ::= \text{a real number (NRf data)} \]
\[ \text{limits: 0:59} \]
\[ <\text{bound}> ::= \text{MAX|MIN} \]
\[ <\text{second}> ::= [ <\text{number}| <\text{bound}> ] \]
\[ <\text{number}> ::= \text{a real number (NRf data)} \]
\[ \text{limits: 0:60} \]
\[ <\text{bound}> ::= \text{MAX|MIN} \]

Example Statements:

\[ \text{OUTPUT 711; "system:time 15, 5, 0"} \]
\[ \text{OUTPUT 711; "SYST:TIME 9, 30, 0"} \]

Query Syntax:

\[ \text{SYSTeM:TIME?} \]

Return Format:

\[ \text{Integer, Integer, Integer} \]

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:CLea
```

**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**

Executes the long confidence test.

**Command Syntax:**

```plaintext
TEST: LONG
```

**Example Statements:**

```plaintext
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

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.
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](image)

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}, <DATA>
```

- `<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> ...
- ::= <integer>
- ::= <floating-point numeric>

Example Statements:

```
OUTPUT 711;"TRAC:DATA D1, TRAC2"
OUTPUT 711;"trac d3, d5"
```

Query Syntax:

```
TRACe[:DATA]? {D1|D2|D3|D4|D5|D6|D7|D8}
```

Return Format:

```
<DEF_BLOCK>
```

- `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>
<thead>
<tr>
<th>CALC:FEED</th>
<th>Measurement Data</th>
<th>Type of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTIM:CORR [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>TCAP [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW:COH</td>
<td>Coherence</td>
<td>Real</td>
</tr>
<tr>
<td>XFR:POW:CROS [1,2</td>
<td>1,3</td>
<td>1,4</td>
</tr>
<tr>
<td>XFR:POW:CROS [1,2</td>
<td>1,3</td>
<td>1,4</td>
</tr>
<tr>
<td>XTIM:VOLT:CDF [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW:RAT [2,1</td>
<td>3,1</td>
<td>4,1</td>
</tr>
<tr>
<td>XTIM:VOLT:HIST [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW:LIN [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW:VAR [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XORD:TRACK [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XTIM:VOLT:PDF [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XFR:POW:FUN</td>
<td>RPM Profile</td>
<td>Real</td>
</tr>
<tr>
<td>XTIM:VOLT [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XTIM:VOLT [1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>XTIM:VOLT-WIND [1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

* 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>
<thead>
<tr>
<th>Lines of Resolution</th>
<th>Alias-Protected Data Points</th>
<th>Data Points Not Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0 to 100</td>
<td>101 to 128</td>
</tr>
<tr>
<td>200</td>
<td>0 to 200</td>
<td>201 to 256</td>
</tr>
<tr>
<td>400</td>
<td>0 to 400</td>
<td>401 to 512</td>
</tr>
<tr>
<td>800</td>
<td>0 to 800</td>
<td>801 to 1024</td>
</tr>
</tbody>
</table>

*First point is Point 0.*

<table>
<thead>
<tr>
<th>Lines of Resolution</th>
<th>Alias-Protected Data Points</th>
<th>Data Points Not Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>14 to 114</td>
<td>0 to 13 115 to 127</td>
</tr>
<tr>
<td>200</td>
<td>28 to 228</td>
<td>0 to 27 229 to 255</td>
</tr>
<tr>
<td>400</td>
<td>56 to 456</td>
<td>0 to 55 457 to 511</td>
</tr>
<tr>
<td>800</td>
<td>112 to 912</td>
<td>0 to 111 913 to 1023</td>
</tr>
</tbody>
</table>

*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.
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]?**

Returns the X-axis data for trace displays.

**Query Syntax:**

```plaintext
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:**

```plaintext
<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:**

```plaintext
<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?**

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]?

Returns the Z-axis data for waterfall displays.

**Query Syntax:**

```
TRACe|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 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?.

---

23-9
TRACe:Z:UNIT?

Returns the unit for the z-axis in waterfall displays.

**Query Syntax:**
```
TRACe:Z:UNIT?
```

**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.

![Diagram of TRIGger function](image_url)

Figure 24-1. The Agilent 35670A’s ARM-INIT-TRIG Functions
TRIGger

TRIGger:EXTernal:FILTer[:LPAS][:STATe] command/query

Turns on a low pass filter for the external trigger.

Command Syntax: 

```
TRIGger:EXTer nal:FILT er[: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:FILT er[: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:EXTer nal:RANGe command/query

Selects the range for the external trigger’s level.

**Command Syntax:**

TRIGger:EXTer nal:RANGe HIGH|LOW

**Example Statements:**

OUTPUT 711;"trig:external:rang HIGH"
OUTPUT 711;"Trig:External:Rang LOW"

**Query Syntax:**

TRIGger:EXTer nal: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 +2 V. 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]

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:

```plaintext
TRIGger:LEVel {<number>[<unit>]|<step>|<bound>

<number> ::= a real number (NRf data)
    limits: -9.9e37:9.9e37
<br 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:

```plaintext
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

Selects the source of the trigger event.

Command Syntax:

```
TRIGger: SOURce IMMEDIATE|EXTERNAL|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.

Maximum pre-trigger value $= 8192 \times \text{sample time}$

$$
sample \text{ time}_{\text{FFT}} = \frac{1}{\text{frequency span} \times 256}
$$

$$
sample \text{ time}_{\text{correlation}} = \frac{\text{record length}}{1024}
$$

Maximum post-trigger delay $= 2^{31} \times \text{sample time}$
The maximum post-trigger delay cannot exceed 1 Msec (1 \times 10^6 \text{ 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>
<thead>
<tr>
<th>Resolution (lines)</th>
<th>Maximum Delay Difference (samples)</th>
<th>Maximum Delay Difference (time records)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>6144</td>
<td>3</td>
</tr>
<tr>
<td>400</td>
<td>7168</td>
<td>7</td>
</tr>
<tr>
<td>200</td>
<td>7680</td>
<td>15</td>
</tr>
<tr>
<td>100</td>
<td>7936</td>
<td>31</td>
</tr>
</tbody>
</table>

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

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:LEVEL command/query

Specifies the level of the tachometer’s input signal which causes the analyzer to trigger.

Command Syntax:

```
TRIGger:TACHometer:LEVEL {<number>[<unit>]|<step>|<bound>
```

- `<number>` ::= a real number (NRf data)
  limits: ±20.0 ±20.0 V (if TRIG:TACH:RANG HIGH)
  ±4.0 ±4.0 V (if TRIG:TACH:RANG LOW)

- `<unit>` ::= [V]

- `<step>` ::= UP|DOWN

- `<bound>` ::= MAX|MIN

Example Statements:

```
OUTPUT 711; "trig:tach:lev 300mv"
OUTPUT711; "TRIGGER:TACHOMETER:LEVEL 2.5"
```

Query Syntax:

```
TRIGGER:TACHOMETER:LEVEL?
```

Return Format:

Real

Attribute Summary:

- Option: not applicable
- Synchronization Required: no
- Preset State: +0.0
- SCPI Compliance: instrument-specific

Description:

The ±4 volt range has 0.0393 V resolution. The ±20 volt range has 0.197 V resolution.

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

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

Specifies the input range of the analyzer’s tachometer.

Command Syntax:

\[ \text{TRIGger:TACHometer:RANGE \{HIGH|LOW\}} \]

Example Statements:

\[ \text{OUTPUT 711;"TRIG:TACH:RANGE HIGH"} \]
\[ \text{OUTPUT 711;"trig:tachometer:rang HIGH"} \]

Query Syntax:

\[ \text{TRIGger:TACHometer:RANGE?} \]

Return Format:

\[ \text{HIGH|LOW} \]

Attribute Summary:

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

Description:

To specify the input range as ±4 volts, send LOW.

To specify the input range as ±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

TRIGger:TACHometer[:RPM]?

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**

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
Introduction

This appendix contains 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

<table>
<thead>
<tr>
<th>Command Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?</td>
<td>query only</td>
<td>Calibrates the analyzer and returns the result</td>
</tr>
<tr>
<td>*CLS</td>
<td>command only</td>
<td>Clears the Status Byte by emptying the error queue and clearing all event registers</td>
</tr>
<tr>
<td>*ESE</td>
<td>‡</td>
<td>Sets bits in the Standard Event enable register</td>
</tr>
<tr>
<td>*ESR?</td>
<td>query only</td>
<td>Reads and clears the Standard Event event register</td>
</tr>
<tr>
<td>*IDN?</td>
<td>query only</td>
<td>Returns a string that uniquely identifies the analyzer</td>
</tr>
<tr>
<td>*OPC</td>
<td>‡</td>
<td>Sets or queries completion of all pending overlapped commands</td>
</tr>
<tr>
<td>*OPT?</td>
<td>query only</td>
<td>Returns a string that identifies the analyzer’s option configuration</td>
</tr>
<tr>
<td>*PCB</td>
<td>command only</td>
<td>Sets the pass-control-back address</td>
</tr>
<tr>
<td>*PSC</td>
<td>‡</td>
<td>Sets the state of the Power-on Status Clear flag</td>
</tr>
<tr>
<td>*RST</td>
<td>command only</td>
<td>Executes a device reset</td>
</tr>
<tr>
<td>*SRE</td>
<td>‡</td>
<td>Sets bits in the Service Request enable register</td>
</tr>
<tr>
<td>*STB?</td>
<td>query only</td>
<td>Reads the Status Byte register</td>
</tr>
<tr>
<td>*TRG</td>
<td>command only</td>
<td>Triggers the analyzer when TRIG:SOUR is BUS</td>
</tr>
<tr>
<td>*TST?</td>
<td>query only</td>
<td>Tests the analyzer hardware and returns the results</td>
</tr>
<tr>
<td>*WAI</td>
<td>command only</td>
<td>Holds off processing of subsequent commands until all preceding commands have been processed</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORt</td>
<td>command only</td>
<td>Stops the current measurement in progress</td>
</tr>
<tr>
<td>ARM</td>
<td>command only</td>
<td>Arm the trigger if ARM:SOUR is MAN</td>
</tr>
<tr>
<td>ARM[:IMMediate]</td>
<td>command only</td>
<td>Specifies the number of RPM in a step for RPM step arming</td>
</tr>
<tr>
<td>ARM:RPM:MODE</td>
<td>‡</td>
<td>Enables the Start RPM Arming qualifier</td>
</tr>
<tr>
<td>ARM:RPM:THReshold</td>
<td>‡</td>
<td>Specifies the starting RPM value</td>
</tr>
<tr>
<td>ARM:SOURce</td>
<td>‡</td>
<td>Specifies the type of arming for the analyzer’s trigger</td>
</tr>
<tr>
<td>ARM:TIMer</td>
<td>‡</td>
<td>Specifies the size of the step used in time step arming</td>
</tr>
<tr>
<td>CALCulate</td>
<td>‡</td>
<td>Selects the active trace(s)</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add ”?” for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
### Agilent 35670A Command Summary

<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CALC[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
## Subsystem Commands Form Description

<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CALC[1 2 3 4]:UNIT:X:USER:TIME:LABel</code></td>
<td>↑</td>
<td>Assigns a name to the user-defined X-axis units in the time domain</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:COUNt</code></td>
<td>↑</td>
<td>Specifies the number of traces stored for waterfall displays</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:DATA?</code> query only</td>
<td></td>
<td>Returns waterfall data that has been transformed to the currently selected coordinate transform (specified with <code>CALC:FORMat</code>)</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:SLICe:COPY</code> command only</td>
<td></td>
<td>Copies the selected waterfall slice to the designated data register</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:SLICe:SELeCT</code> ↑</td>
<td></td>
<td>Selects the waterfall slice at the specified X-axis position</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:SLICe:SELeCT:POINt</code> ↑</td>
<td></td>
<td>Selects a waterfall slice by its display point value</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:TRACe:COPY</code> command only</td>
<td></td>
<td>Saves the trace to the specified data register</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WATerfall:TRACe:SELeCT</code> ↑</td>
<td></td>
<td>Selects a waterfall trace by its Z-axis value</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:WAT:TRACe:SELeCT:POINt</code> ↑</td>
<td></td>
<td>Selects a waterfall trace by its step value</td>
</tr>
<tr>
<td><code>CALC[1 2 3 4]:X:DATA?</code> query only</td>
<td></td>
<td>Returns X-axis values that correspond to the Y-axis values read with the <code>CALC:DATA?</code> query</td>
</tr>
</tbody>
</table>

### CALibration

| CALibration[:ALL]? query only | query only | Calibrates the analyzer and returns the result |
| CALibration:AUTO ↑ | | Calibrates the analyzer or sets the state of the autocalibration function |

### DISPLAY

| 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
## Agilent 35670A Command Summary

<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPlay:RPM[:STATe]</td>
<td>‡</td>
<td>Turns on the display of the RPM indicator</td>
</tr>
<tr>
<td>DISPlay:STATe</td>
<td>‡</td>
<td>Turns on (or turns off) the analyzer’s display</td>
</tr>
<tr>
<td>DISPlay:TCAPture:ENVelope[:STATe]</td>
<td>‡</td>
<td>Displays the envelope of the time capture buffer</td>
</tr>
<tr>
<td>DISPlay:VIEW</td>
<td>‡</td>
<td>Specifies what is displayed on the analyzer’s screen</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISPlay[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISPlay[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISPlay[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
### Subsystem Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DISP[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

### FORMat

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMat[:DATA]</td>
<td>‡</td>
<td>Specifies the data type and date encoding to be used during transfers of a data block</td>
</tr>
</tbody>
</table>

### HCOPY

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOPY:COLor:DEFault</td>
<td>command only</td>
<td>Specifies default values for the plotter pen assignments</td>
</tr>
<tr>
<td>HCOPY:DESTination</td>
<td>‡</td>
<td>Specifies where the print or plot operation is sent: either to a device or to a file on the default disk</td>
</tr>
<tr>
<td>HCOPY:DEVice:LANGuage</td>
<td>‡</td>
<td>Specifies the format of the plot/print output</td>
</tr>
<tr>
<td>HCOPY:DEVice:SPEed</td>
<td>‡</td>
<td>Specifies the plotting speed for all plotting operations initiated by the analyzer</td>
</tr>
<tr>
<td>HCOPY[:IMMediate]</td>
<td>command only</td>
<td>Plots or prints the currently specified item</td>
</tr>
<tr>
<td>HCOPY[:ITEM:ALL[:IMMediate]]</td>
<td>command only</td>
<td>Plots or prints the entire screen</td>
</tr>
<tr>
<td>HCOPY[:ITEM:FFEed:STATe]</td>
<td>‡</td>
<td>Turns the page-eject feature on or off</td>
</tr>
<tr>
<td>HCOPY[:ITEM:LABel:COLor]</td>
<td>‡</td>
<td>Selects the pen used for plotting miscellaneous annotations</td>
</tr>
<tr>
<td>HCOPY[:ITEM:LABel:TEXT]</td>
<td>‡</td>
<td>Prints a label on the plot and print output</td>
</tr>
<tr>
<td>HCOPY[:ITEM:LABel:TEXT]</td>
<td>‡</td>
<td>Specifies a label for plot and print output</td>
</tr>
<tr>
<td>HCOPY[:ITEM:TDSTamp:FORMat]</td>
<td>‡</td>
<td>Specifies the format of the time stamp used for plotting and printing</td>
</tr>
<tr>
<td>HCOPY[:ITEM:TDSTamp:STATe]</td>
<td>‡</td>
<td>Turns a time stamp on or off for print and plot operations</td>
</tr>
<tr>
<td>HCOPY[:ITEM:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add ‘?’ for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>:GRATicule:COlor</td>
<td>‡</td>
<td>Selects the pen used to plot the trace graticules</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>:GRATicule[:IMMediate]</td>
<td></td>
<td>Plots or prints the currently displayed trace(s)</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WIND[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:ITEM[:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HCOPy:PAGE:DIMensions:AUto</td>
<td>‡</td>
<td>Specifies P1 and P2 values for a plotter</td>
</tr>
<tr>
<td>HCOPy:PAGE:DIMensions:USER:LLEFt</td>
<td>‡</td>
<td>Specifies the lower left position (P1) of the plot area</td>
</tr>
<tr>
<td>HCOPy:PAGE:DIMensions:USER:URIGht</td>
<td>‡</td>
<td>Specifies the upper right position (P2) of the plot area</td>
</tr>
<tr>
<td>HCOPy:PLOT:ADDRess</td>
<td>‡</td>
<td>Tells the analyzer which GPIB address is assigned to your plotter</td>
</tr>
<tr>
<td>HCOPy:PRINt:ADDRess</td>
<td>‡</td>
<td>Tells the analyzer which GPIB address is assigned to your printer</td>
</tr>
<tr>
<td>HCOPy:TITLe[1</td>
<td>2]</td>
<td>‡</td>
</tr>
</tbody>
</table>

INITiate

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 |

INPut

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
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPut[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>INPut[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>INPut[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**INSTrument**

| INSTrument:NSELect | ‡ | Selects one of the analyzer’s six major instrument modes |
| INSTrument[:SELect] | ‡ | Selects one of the analyzer’s six major instrument modes |

**MEMory**

| 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 |

**MMEMory**

| 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:FSYSstem? | 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:DATAble: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
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMEMory:LOAD:MATH</td>
<td>command only</td>
<td>Loads a complete set of math definitions into the analyzer from the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:PROGram</td>
<td>command only</td>
<td>Loads an Instrument BASIC program into the analyzer from a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:STATe</td>
<td>command only</td>
<td>Loads an instrument state into the analyzer from the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:SYNthesis</td>
<td>command only</td>
<td>Loads a synthesis table into the analyzer from a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:TCAPture</td>
<td>command only</td>
<td>Loads a time capture file from the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:TRACe</td>
<td>command only</td>
<td>Loads a trace into the analyzer from the specified disk</td>
</tr>
<tr>
<td>MMEMory:LOAD:WATerfall</td>
<td>command only</td>
<td>Loads a waterfall file into the analyzer from a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:MDIrectory</td>
<td>command only</td>
<td>@RSUB1 = Command Syntax:</td>
</tr>
<tr>
<td>MMEMory:MOVE</td>
<td>command only</td>
<td>Renames a file</td>
</tr>
<tr>
<td>MMEMory:MSIS</td>
<td>‡</td>
<td>Specifies a default disk</td>
</tr>
<tr>
<td>MMEMory:NAME</td>
<td>‡</td>
<td>Specifies a filename for the output of a print or plot operation</td>
</tr>
<tr>
<td>MMEMory:STORe:CFIT</td>
<td>command only</td>
<td>Stores a curve fit table to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:CONTinue</td>
<td>‡</td>
<td>Splits a large file (a time capture file or a waterfall file) over multiple disks</td>
</tr>
<tr>
<td>MMEMory:STORe:DTAble:TRACe[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MMEM:STORe:LIMit:LOWer:TRACe[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MMEM:STORe:LIMit:UPPer:TRACe[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MMEMory:STORe:MATH</td>
<td>command only</td>
<td>Saves a complete set of math definitions to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:PROGram</td>
<td>command only</td>
<td>Saves an Instrument BASIC program to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:PROGram:FORMat</td>
<td>‡</td>
<td>Specifies the format Agilent Instrument BASIC programs are stored</td>
</tr>
<tr>
<td>MMEMory:STORe:STATe</td>
<td>command only</td>
<td>Saves the instrument state to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:SYNthesis</td>
<td>command only</td>
<td>Stores a synthesis table to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:TCAPture</td>
<td>command only</td>
<td>Saves the time capture buffer to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:TRACe</td>
<td>command only</td>
<td>Saves the specified trace to a file on the specified disk</td>
</tr>
<tr>
<td>MMEMory:STORe:WATerfall</td>
<td>command only</td>
<td>Saves the current waterfall display to a file on the specified disk</td>
</tr>
</tbody>
</table>

**OUTPut**

| OUTPut:FILTer[:LPASs][:STATe] | ‡ | Turns on a low pass filter for the arbitrary source |

‡ Command form; add “?” for query form
### Subsystem Commands Form Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPut[:STATe]</td>
<td>‡</td>
<td>Enables the analyzer’s internal source</td>
</tr>
</tbody>
</table>

#### PROGram

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGram:EDIT:ENABLE</td>
<td>‡</td>
<td>Disables the Agilent Instrument BASIC editor</td>
</tr>
<tr>
<td>PROGram:EXPLICIT:DEFINE</td>
<td>‡</td>
<td>Loads an Agilent Instrument BASIC program into the specified program buffer from an external controller</td>
</tr>
<tr>
<td>PROGram:EXPLICIT:LABEL</td>
<td>‡</td>
<td>Loads a softkey label for the specified HP Instrument BASIC program</td>
</tr>
</tbody>
</table>

#### HP Instrument BASIC program

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGram[:SELECTed]:DEFINE</td>
<td>‡</td>
<td>Loads an HP Instrument BASIC program from an external controller into the active program buffer</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:DELETE:ALL</td>
<td>command only</td>
<td>Deletes all HP Instrument BASIC programs stored in the analyzer</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:DELETE[:SELECTed]</td>
<td>command only</td>
<td>Deletes the active HP Instrument BASIC program</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:LABEL</td>
<td>‡</td>
<td>Loads a softkey label for the active HP Instrument BASIC program</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:MALLOCATE</td>
<td>‡</td>
<td>Allocates memory space for HP Instrument BASIC programs</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:NAME</td>
<td>‡</td>
<td>Selects an HP Instrument BASIC program</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:NUMBER</td>
<td>‡</td>
<td>Loads a new value for the specified numeric variable in the active HP Instrument BASIC program</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:STATE</td>
<td>‡</td>
<td>Selects the state of the active HP Instrument BASIC program</td>
</tr>
<tr>
<td>PROGram[:SELECTed]:STRING</td>
<td>‡</td>
<td>Loads a new value for the specified string variable for the active HP Instrument BASIC program</td>
</tr>
</tbody>
</table>

#### [:SENSe]

<table>
<thead>
<tr>
<th>Command</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SENSe]:AVERAGE:CONFidence</td>
<td>‡</td>
<td>Specifies the confidence level used in equal confidence averaging in octave measurements</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:COUNT</td>
<td>‡</td>
<td>Specifies a count or a weighting factor for the averaged measurement data</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:HOLD</td>
<td>‡</td>
<td>Specifies the type of hold used in averaging octave measurements</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:IMPulse</td>
<td>‡</td>
<td>Enables impulse detection in octave measurements</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:RESULT:RATE</td>
<td>‡</td>
<td>Specifies how often the display is updated when fast average mode is on</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:RESULT[:STATE]</td>
<td>‡</td>
<td>Selects fast average mode</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:PREVIEW</td>
<td>‡</td>
<td>Specifies the type of preview averaging</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:PREVIEW:ACCEPT</td>
<td>command only</td>
<td>Accept the current time record during preview averaging</td>
</tr>
<tr>
<td>[SENSe]:AVERAGE:PREVIEW:REJECT</td>
<td>command only</td>
<td>Reject the current time record during preview averaging</td>
</tr>
</tbody>
</table>

‡ Command form; add “?“ for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SENSe:]AVERage:PREView:TIME</td>
<td>‡</td>
<td>Specifies the amount of time the analyzer waits for a response in timed preview averaging</td>
</tr>
<tr>
<td>[SENSe:]AVERage:[STATE]</td>
<td>‡</td>
<td>Turns the selected averaging function (AVER:TYPE) on or off</td>
</tr>
<tr>
<td>[SENSe:]AVERage:TCONtrol</td>
<td>‡</td>
<td>Specifies how the analyzer behaves after the count (AVER:COUN) is reached</td>
</tr>
<tr>
<td>[SENSe:]AVERage:TIME</td>
<td>‡</td>
<td>Specify the time period used in averaging octave measurements and histograms</td>
</tr>
<tr>
<td>[SENSe:]AVERage:TYPE</td>
<td>‡</td>
<td>Specifies the type of averaging the analyzer performs</td>
</tr>
<tr>
<td>[SENSe:]FEED</td>
<td>‡</td>
<td>Specifies the data source for a measurement; either from the input channels or from the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:BLOCksize</td>
<td>‡</td>
<td>Specifies the number of real-time data points displayed on the analyzer’s screen</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:CENTer</td>
<td>‡</td>
<td>Specifies the center frequency for the current measurement</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:MANual</td>
<td>‡</td>
<td>Selects a discrete point to be measured during manual sweep mode</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:RESolution</td>
<td>‡</td>
<td>Specifies the frequency measurement resolution for FFT and swept sine instrument modes</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:RESolution:AUTO</td>
<td>‡</td>
<td>Selects auto resolution for swept sine instrument mode</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:RESolution:AUTO:MCHange</td>
<td>‡</td>
<td>Specifies the maximum change permitted between the frequency response of the current measurement point and the frequency response of the previous measurement point</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:RESolution:AUTO:MINimum</td>
<td>‡</td>
<td>Specifies the initial resolution of a swept sine measurement with automatic resolution</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:RESolution:OCTave</td>
<td>‡</td>
<td>Specifies the type of octave measurement</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:SPAN</td>
<td>‡</td>
<td>Specifies the frequency bandwidth to be measured</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:SPAN:FULL</td>
<td>command only</td>
<td>Sets the analyzer to the widest frequency span available for the current instrument mode</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:SPAN:LINK</td>
<td>‡</td>
<td>Specifies the frequency parameter which remains constant if frequency span or record length is modified</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:START</td>
<td>‡</td>
<td>Specifies the start (lowest) frequency for the frequency band of the current measurement</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:STEP[:INCRement]</td>
<td>‡</td>
<td>Specifies the step size which is used when changing frequency parameters</td>
</tr>
<tr>
<td>[SENSe:]FREQuency:STOP</td>
<td>‡</td>
<td>Sets the stop frequency to the specified value</td>
</tr>
<tr>
<td>[SENSe:]HISTogram:BINS</td>
<td>‡</td>
<td>Specifies the number of bins in a histogram</td>
</tr>
<tr>
<td>[SENSe:]ORDer:MAXimum</td>
<td>‡</td>
<td>Specifies the number of orders to be displayed</td>
</tr>
<tr>
<td>[SENSe:]ORDer:RESolution</td>
<td>‡</td>
<td>Specifies order resolution</td>
</tr>
<tr>
<td>[SENSe:]ORDer:RESolution:TRACK</td>
<td>‡</td>
<td>Specifies the number of points per order track</td>
</tr>
<tr>
<td>[SENSe:]ORDer:RPM:MAXimum</td>
<td>‡</td>
<td>Specifies the maximum rotational speed range you want to analyze</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
### Subsystem Commands Form Description

<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SENSe:]ORDer:RPM:MINimum ‡</td>
<td>‡</td>
<td>Specifies the minimum rotational speed range you want to analyze</td>
</tr>
<tr>
<td>[SENSe:]ORDer:TRACK{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]ORDer:TRACK{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]REFe rence ‡</td>
<td>‡</td>
<td>Specifies the reference channel(s)</td>
</tr>
<tr>
<td>[SENSe:]REject:STAtE ‡</td>
<td>‡</td>
<td>Turns overload rejection on or off</td>
</tr>
<tr>
<td>[SENSe:]SWEep:DIRection ‡</td>
<td>‡</td>
<td>Specifies the direction of the sweep</td>
</tr>
<tr>
<td>[SENSe:]SWEep:DWELl ‡</td>
<td>‡</td>
<td>Specifies the integration time for swept sine measurements</td>
</tr>
<tr>
<td>[SENSe:]SWEep:MODE ‡</td>
<td>‡</td>
<td>Specifies automatic or manual sweep modes</td>
</tr>
<tr>
<td>[SENSe:]SWEep:OVERlap ‡</td>
<td>‡</td>
<td>Specifies the maximum amount of time record overlap</td>
</tr>
<tr>
<td>[SENSe:]SWEep:SPACing ‡</td>
<td>‡</td>
<td>Selects linear or logarithmic spacing between measurement data points</td>
</tr>
<tr>
<td>[SENSe:]SWEep:STIMe ‡</td>
<td>‡</td>
<td>Specifies the settling time for a swept sine measurement</td>
</tr>
<tr>
<td>[SENSe:]SWEep:TIME ‡</td>
<td>‡</td>
<td>Specifies the length of the time record in seconds</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:ABORt command only</td>
<td>command only</td>
<td>Stops the time capture process</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:DELe te command only</td>
<td>command only</td>
<td>Removes the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]TCAPture[:IMMediate] command only</td>
<td>command only</td>
<td>Starts the collection of data for the time capture process</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:LENGTH ‡</td>
<td>‡</td>
<td>Specifies the length of the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:MA LLocate command only</td>
<td>command only</td>
<td>Allocates memory for the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:STARt{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:STOP{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:TACHometer:RPM:MAXimum ‡</td>
<td>‡</td>
<td>Specifies the tachometer’s maximum RPM when included in the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]TCAPture:TACHometer[:STAtE] ‡</td>
<td>‡</td>
<td>Directs the analyzer to include the tachometer input signal in the time capture buffer</td>
</tr>
<tr>
<td>[SENSe:]VOLTage{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]VOLTage{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]VOLTage{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]VOLTage{1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENSe:]VOLTage{1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SENS]:VOLTage[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENS]:VOLT[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENS]:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENS]:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENS]:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[SENS]:WINDow[1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**SOURce**

<table>
<thead>
<tr>
<th>SOURce</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURce:BUrSt</td>
<td>‡</td>
<td>Sets the burst length for the burst source types</td>
</tr>
<tr>
<td>SOURce:FREQuency[:CW]</td>
<td>‡</td>
<td>Sets the frequency of the sine source</td>
</tr>
<tr>
<td>SOURce:FREQuency:FIXed</td>
<td>‡</td>
<td>Sets the frequency of the sine source type</td>
</tr>
<tr>
<td>SOURce:FUNCtion[:SHAPe]</td>
<td>‡</td>
<td>Specifies the source output</td>
</tr>
<tr>
<td>SOURce:USER:CAPTure</td>
<td>‡</td>
<td>Specifies which channel of time capture data to use</td>
</tr>
<tr>
<td>SOURce:USER[:REGister]</td>
<td>‡</td>
<td>Specifies the data register which contains the data for the arbitrary source</td>
</tr>
<tr>
<td>SOURce:USER:REPeat</td>
<td>‡</td>
<td>Enables the source repeat function</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel]:AUTO</td>
<td>‡</td>
<td>Enables the autolevel feature in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]</td>
<td>‡</td>
<td>Specifies the source output level</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet</td>
<td>‡</td>
<td>Specifies a DC offset for the source output</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel]:REFerence</td>
<td>‡</td>
<td>Specifies the amplitude of the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel]:REFerence:CHANnel</td>
<td>‡</td>
<td>Selects the reference input channel for the autolevel feature in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage[:LEVel]:REFerence:TOLerance</td>
<td>‡</td>
<td>Specifies the sensitivity of the autolevel algorithm in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage:LIMit[:AMPLitude]</td>
<td>‡</td>
<td>Sets the maximum limit used by the autolevel algorithm to adjust the source’s amplitude in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage:LIMit:INPut</td>
<td>‡</td>
<td>Sets the maximum amplitude of the non-reference input channels for the autolevel feature in swept sine measurements (INST:SEL SINE)</td>
</tr>
<tr>
<td>SOURce:VOLTage:SLEW</td>
<td>‡</td>
<td>Specifies the source amplitude ramp rate in swept sine measurements (INST:SEL SINE)</td>
</tr>
</tbody>
</table>

**STATus**

‡ Command form; add “?” for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATus:DEVice:CONDition?</td>
<td>query only</td>
<td>Reads and clears the Device State condition register</td>
</tr>
<tr>
<td>STATus:DEVice:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the Device State enable register</td>
</tr>
<tr>
<td>STATus:DEVice:EVENt?</td>
<td>query only</td>
<td>Reads and clears the Device State event register</td>
</tr>
<tr>
<td>STATus:DEVice:NTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Device Status negative transition register</td>
</tr>
<tr>
<td>STATus:DEVice:PTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Device Status positive transition register</td>
</tr>
<tr>
<td>STATus:OPERation:CONDition?</td>
<td>query only</td>
<td>Reads the Operation Status condition register</td>
</tr>
<tr>
<td>STATus:OPERation:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the Operation Status enable register</td>
</tr>
<tr>
<td>STATus:OPERation:EVENt?</td>
<td>query only</td>
<td>Reads and clears the Operation Status event register</td>
</tr>
<tr>
<td>STATus:OPERation:NTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Operation Status negative transition register</td>
</tr>
<tr>
<td>STATus:OPERation:PTRansition</td>
<td>‡</td>
<td>Sets bits in the Operation Status positive transition register</td>
</tr>
<tr>
<td>STATus:PRESet</td>
<td>command only</td>
<td>Sets bits in most enable and transition registers to their default state</td>
</tr>
<tr>
<td>STATus:QUEStionable:CONDition?</td>
<td>query only</td>
<td>Reads and clears the Questionable Status condition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the Questionable Status enable register</td>
</tr>
<tr>
<td>STATus:QUEStionable:EVENt?</td>
<td>query only</td>
<td>Reads and clears the Questionable Status event register</td>
</tr>
<tr>
<td>STATus:QUEStionable:LIMit:CONDition?</td>
<td>query only</td>
<td>Reads and clears the Limit Fail condition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:LIMit:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the Limit Fail enable register</td>
</tr>
<tr>
<td>STATus:QUEStionable:LIMit:EVENt?</td>
<td>query only</td>
<td>Reads and clears the Limit Fail event register</td>
</tr>
<tr>
<td>STATus:QUEStionable:LIMit:NTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Limit Fail negative transition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:LIMit:PTRansition</td>
<td>‡</td>
<td>Sets queries bits in the Limit Fail positive transition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:NTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Questionable Status negative transition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:PTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Questionable Status positive transition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:VOLTage:CONDition?</td>
<td>query only</td>
<td>Reads the Questionable Voltage condition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:VOLTage:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the Questionable Voltage enable register</td>
</tr>
<tr>
<td>STATus:QUEStionable:VOLTage:EVENt?</td>
<td>query only</td>
<td>Reads and clears the Questionable Voltage event register</td>
</tr>
<tr>
<td>STATus:QUEStionable:VOLTage:NTRansition</td>
<td>‡</td>
<td>Sets and queries bits in the Questionable Voltage negative transition register</td>
</tr>
<tr>
<td>STATus:QUEStionable:VOLTage:PTRansition</td>
<td>‡</td>
<td>Sets bits in the Questionable Voltage positive transition register</td>
</tr>
<tr>
<td>STATus:USER:ENABle</td>
<td>‡</td>
<td>Sets and queries bits in the User Defined enable register</td>
</tr>
<tr>
<td>STATus:USER:EVENt?</td>
<td>query only</td>
<td>Reads and clears the User Defined event register</td>
</tr>
</tbody>
</table>

‡ Command form; add ‘?’ for query form
<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATus:USER:PULSe</td>
<td>command only</td>
<td>Sets bits in the User Defined event register</td>
</tr>
<tr>
<td>SYSTem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTem:BEEPer[:IMMediate]</td>
<td>command only</td>
<td>Generates the tone at a given frequency from the analyzer’s beeper</td>
</tr>
<tr>
<td>SYSTem:BEEPer:STATE</td>
<td>‡</td>
<td>Enables the analyzer’s beeper</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:GPIB[:SELF]:ADDReSS</td>
<td>‡</td>
<td>Sets the analyzer’s GPIB address</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:BAUD</td>
<td>‡</td>
<td>Specifies the data-transfer rate between the analyzer and RS-232-C peripheral devices</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:BITS</td>
<td>‡</td>
<td>Specifies the number of bits in a character for the RS-232-C interface</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:PACE</td>
<td>‡</td>
<td>Sets the RS-232-C receiver handshake pacing type</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:PARity:CHECK</td>
<td>‡</td>
<td>Enables RS-232-C parity verification capability</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:PARity[:TYPE]</td>
<td>‡</td>
<td>Sets the parity generated for characters transmitted over the RS-232-C interface</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial[:RECeive]:SBITs</td>
<td>‡</td>
<td>Specifies the number of “stop bits” sent with each character over the RS-232-C interface</td>
</tr>
<tr>
<td>SYSTem:COMMUnicate:SERial:TRANsmit:PACE</td>
<td>‡</td>
<td>Sets the RS-232 transmitter handshake pacing type</td>
</tr>
<tr>
<td>SYSTem:DATE</td>
<td>‡</td>
<td>Sets the date in the analyzer’s battery-backed clock</td>
</tr>
<tr>
<td>SYSTem:ERROR?</td>
<td>query only</td>
<td>Returns one error message from the analyzer’s error queue</td>
</tr>
<tr>
<td>SYSTem:FAN[:STATe]</td>
<td>‡</td>
<td>Sets the fan’s operating mode</td>
</tr>
<tr>
<td>SYSTem:FLOG:CLEar</td>
<td>command only</td>
<td>Clears the fault log of all entries</td>
</tr>
<tr>
<td>SYSTem:KEY</td>
<td>‡</td>
<td>Writes or queries front-panel key presses</td>
</tr>
<tr>
<td>SYSTem:KLOCk</td>
<td>‡</td>
<td>Disables the keyboard</td>
</tr>
<tr>
<td>SYSTem:POWer:SOURce?</td>
<td>query only</td>
<td>Queries the setting of the rear-panel power select switch</td>
</tr>
<tr>
<td>SYSTem:POWer:STATe</td>
<td>‡</td>
<td>Turns off the analyzer’s power</td>
</tr>
<tr>
<td>SYSTem:PRESet</td>
<td>command only</td>
<td>Returns most of the analyzer’s parameters to their preset states</td>
</tr>
<tr>
<td>SYSTem:SET</td>
<td>‡</td>
<td>Transfers an instrument state between the analyzer and an external controller</td>
</tr>
<tr>
<td>SYSTem:TIME</td>
<td>‡</td>
<td>Sets the time in the analyzer’s battery-backed clock</td>
</tr>
<tr>
<td>SYSTem:VERSion?</td>
<td>query only</td>
<td>Returns the SCPI version to which the analyzer complies</td>
</tr>
<tr>
<td>TEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST:LOG:CLEar</td>
<td>command only</td>
<td>Clears the test log</td>
</tr>
<tr>
<td>TEST:LONG</td>
<td>command only</td>
<td>Executes the long confidence test</td>
</tr>
<tr>
<td>TEST:LONG:RESUlt?</td>
<td>query only</td>
<td>Returns the overall result of the long confidence test</td>
</tr>
</tbody>
</table>

‡ Command form; add “?” for query form
### Agilent 35670A Command Summary

<table>
<thead>
<tr>
<th>Subsystem Commands</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRACe</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACe[:DATA]</td>
<td>‡</td>
<td>Stores data to the specified data register</td>
</tr>
<tr>
<td>TRACe:WATerfall[:DATA]</td>
<td>‡</td>
<td>Stores data to the specified waterfall register</td>
</tr>
<tr>
<td>TRACe:X[:DATA]?</td>
<td>query only</td>
<td>Returns the X-axis data for trace displays</td>
</tr>
<tr>
<td>TRACe:X:UNIT?</td>
<td>query only</td>
<td>Returns the unit for the x-axis for trace displays</td>
</tr>
<tr>
<td>TRACe:Z[:DATA]?</td>
<td>query only</td>
<td>Returns the Z-axis data for waterfall displays</td>
</tr>
<tr>
<td>TRACe:Z:UNIT?</td>
<td>query only</td>
<td>Returns the unit for the z-axis in waterfall displays</td>
</tr>
<tr>
<td><strong>TRIGger</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIGger:EXThernal:FILTer[:LPAS][:STATe]</td>
<td>‡</td>
<td>Turns on a low pass filter for the external trigger</td>
</tr>
<tr>
<td>TRIGger:EXThernal:LEVel</td>
<td>‡</td>
<td>Specifies the level of the external input signal which causes the analyzer to trigger</td>
</tr>
<tr>
<td>TRIGger:EXThernal:RANGe</td>
<td>‡</td>
<td>Selects the range for the external trigger’s level</td>
</tr>
<tr>
<td>TRIGger[:IMMediate]</td>
<td>command only</td>
<td>Triggers the analyzer if TRIG:SOUR is BUS</td>
</tr>
<tr>
<td>TRIGger:LEVEL</td>
<td>‡</td>
<td>Specifies the level of the input signal which causes the analyzer to trigger</td>
</tr>
<tr>
<td>TRIGger:SLOPe</td>
<td>‡</td>
<td>Specifies the slope of the signal which triggers the analyzer</td>
</tr>
<tr>
<td>TRIGger:SOURce</td>
<td>‡</td>
<td>Selects the source of the trigger event</td>
</tr>
<tr>
<td>TRIGger:STARt[1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>TRIGger:TACHometer:HOLDoff</td>
<td>‡</td>
<td>Specifies a time interval in which the tachometer trigger is inhibited</td>
</tr>
<tr>
<td>TRIGger:TACHometer:LEVel</td>
<td>‡</td>
<td>Specifies the level of the tachometer’s input signal which causes the analyzer to trigger</td>
</tr>
<tr>
<td>TRIGger:TACHometer:PCOunt</td>
<td>‡</td>
<td>Specifies the number of tachometer pulses that occur in one revolution of the shaft</td>
</tr>
<tr>
<td>TRIGger:TACHometer:RANGe</td>
<td>‡</td>
<td>Specifies the input range of the analyzer’s tachometer</td>
</tr>
<tr>
<td>TRIGger:TACHometer[:RPM]?</td>
<td>query only</td>
<td>Reads and returns the current RPM value for the analyzer’s tachometer</td>
</tr>
<tr>
<td>TRIGger:TACHometer:SLOPe</td>
<td>‡</td>
<td>Specifies the slope of the tachometer input signal to be used in RPM step arming or order tracking</td>
</tr>
</tbody>
</table>

‡ Command form; add "?" for query form
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.
# Cross-Reference from Front-Panel Keys to GPIB Commands

## Measurement Group

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inst Mode</strong></td>
<td></td>
</tr>
<tr>
<td>[ FFT ANALYSIS ]</td>
<td>INSTRument:SElect FFT</td>
</tr>
<tr>
<td>[ OCTAVE ANALYSIS ]</td>
<td>INSTRument:SElect OCTave</td>
</tr>
<tr>
<td>[ ORDER ANALYSIS ]</td>
<td>INSTRument:SElect ORDER</td>
</tr>
<tr>
<td>[ SWEPT SINE ]</td>
<td>INSTRument:SElect SINE</td>
</tr>
<tr>
<td>[ CORRELATN ANALYSIS ]</td>
<td>INSTRument:SElect CORRelation</td>
</tr>
<tr>
<td>[ HISTOGRAM / TIME ]</td>
<td>INSTRument:SElect HISTogram</td>
</tr>
<tr>
<td>[ CAPTURE ON OFF ]</td>
<td>[SENSe:FEED INPut</td>
</tr>
<tr>
<td>[ 1 CHANNEL ]</td>
<td>INPut2[:STATe] OFF</td>
</tr>
<tr>
<td>[ 2 CHANNELS ]</td>
<td>INPut2[:STATe] ON</td>
</tr>
<tr>
<td>[ 4 CHANNELS ]</td>
<td>INPut4[:STATe] ON</td>
</tr>
<tr>
<td>[ REF CHANNELS 1, 3 ]</td>
<td>[SENSe:REference SINGle</td>
</tr>
<tr>
<td>[ TIME CAPTURE ]</td>
<td></td>
</tr>
<tr>
<td>[ MEAS FROM INP BUFR ]</td>
<td>[SENSe:FEED INPut</td>
</tr>
<tr>
<td>[ FILL BUFFER ]</td>
<td>[SENSe:TCAPture:IMMediate]</td>
</tr>
<tr>
<td>[ ABORT FILL ]</td>
<td>[SENSe:TCAPture:ABORt]</td>
</tr>
<tr>
<td>[ BUFFER LENGTH ]</td>
<td>[SENSe:TCAPture:LENGth 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ ALLOCATE BUFFER ]</td>
<td>[SENSe:TCAPture:MALLocate]</td>
</tr>
<tr>
<td>[ CONFIRM ALLOCATE ]</td>
<td></td>
</tr>
<tr>
<td>[ REMOVE CAPTURE]</td>
<td>[SENSe:TCAPture:DELeTe]</td>
</tr>
<tr>
<td>[ CONFIRM REMOVE ]</td>
<td></td>
</tr>
<tr>
<td>[ TACHOMETER OPTIONS ]</td>
<td>[SENSe:TCAPture:TACHometer[:STATe] OFF</td>
</tr>
<tr>
<td>[ MAX RPM ]</td>
<td>[SENSe:TCAPture:TACHometer:RPM:MAXimum 0 ~ 9.9e+37]</td>
</tr>
<tr>
<td>[ ANALYSIS REGION ]</td>
<td></td>
</tr>
<tr>
<td>[ STRT TIME CHANNEL 1 ]</td>
<td>[SENSe:TCAPture:STARt1 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STOP TIME CHANNEL 1 ]</td>
<td>[SENSe:TCAPture:STOP1 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STRT TIME CHANNEL 2 ]</td>
<td>[SENSe:TCAPture:STARt2 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STOP TIME CHANNEL 2 ]</td>
<td>[SENSe:TCAPture:STOP2 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STRT TIME CHANNEL 3 ]</td>
<td>[SENSe:TCAPture:STARt3 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STOP TIME CHANNEL 3 ]</td>
<td>[SENSe:TCAPture:STOP3 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STRT TIME CHANNEL 4 ]</td>
<td>[SENSe:TCAPture:STARt4 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ STOP TIME CHANNEL 4 ]</td>
<td>[SENSe:TCAPture:STOP4 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ ENVELOPE ON OFF ]</td>
<td>DISPly:TCAPture:ENVelope ON OFF</td>
</tr>
</tbody>
</table>
## Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Freq</strong></td>
<td></td>
</tr>
<tr>
<td>[ SPAN ]</td>
<td>(SENSe):FREQuency:SPAN 0.015625 ~ 102400[HZ]</td>
</tr>
<tr>
<td>[ CENTER ]</td>
<td>(SENSe):FREQuency:CENTer 0.0234375 ~ 115000[HZ]</td>
</tr>
<tr>
<td>[ START ]</td>
<td>(SENSe):FREQuency:STARt 0 ~ 115000[HZ]</td>
</tr>
<tr>
<td>[ STOP ]</td>
<td>(SENSe):FREQuency:STOP 0.03125 ~ 115000[HZ]</td>
</tr>
<tr>
<td>[ ZERO START ]</td>
<td>(SENSe):FREQuency:STARt 0</td>
</tr>
<tr>
<td>[ FULL SPAN ]</td>
<td>(SENSe):FREQuency:SPAN:FULL</td>
</tr>
<tr>
<td>[ ENTRY STEP SIZE ]</td>
<td>(SENSe):FREQuency:STEP[:INCRement] 0.015625 ~ 10240</td>
</tr>
<tr>
<td>[ RECORD LENGTH ]</td>
<td>(SENSe):SWEep:TIME 0.000976562 ~ 8192[S]</td>
</tr>
<tr>
<td>[ RESOLUTN LINES ]</td>
<td>(SENSe):FREQuency:RESolution 100</td>
</tr>
<tr>
<td>[ 1/1 OCTAVE ]</td>
<td>(SENSe):FREQuency:RESolution:OCTave FULL</td>
</tr>
<tr>
<td>[ 1/3 OCTAVE ]</td>
<td>(SENSe):FREQuency:RESolution:OCTave THIRd</td>
</tr>
<tr>
<td>[ 1/12 OCTAVE ]</td>
<td>(SENSe):FREQuency:RESolution:OCTave TWELth</td>
</tr>
<tr>
<td>[ MIN RPM ]</td>
<td>(SENSe):ORDer:RPM:MINimum 0 ~ 9.9e + 37</td>
</tr>
<tr>
<td>[ MAX RPM ]</td>
<td>(SENSe):ORDer:RPM:MAXimum 0 ~ 9.9e + 37</td>
</tr>
<tr>
<td>[ MAX ORDER ]</td>
<td>(SENSe):ORDer:MAXimum 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ DELTA ORDER ]</td>
<td>(SENSe):ORDer:RESolution 0.0078125 ~ 1[ORD]</td>
</tr>
<tr>
<td>[ TRACK ON OFF ]</td>
<td>(SENSe):ORDer:TRACk:STATe OFF</td>
</tr>
<tr>
<td>[ TRACK SETUP ]</td>
<td>(SENSe):ORDer:TRACk1 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ TRACK 2 ORDER ]</td>
<td>(SENSe):ORDer:TRACk2 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ TRACK 3 ORDER ]</td>
<td>(SENSe):ORDer:TRACk3 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ TRACK 4 ORDER ]</td>
<td>(SENSe):ORDer:TRACk4 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ TRACK 5 ORDER ]</td>
<td>(SENSe):ORDer:TRACk5 0 ~ 9.9e + 37[ORD]</td>
</tr>
<tr>
<td>[ TRACK POINTS ]</td>
<td>(SENSe):ORDer:RESolution:TRACk 1 ~ 2048</td>
</tr>
<tr>
<td>[ SWEEP UP DOWN ]</td>
<td>(SENSe):SWEep:DIRection UP</td>
</tr>
<tr>
<td>[ SWEEP AUTO MAN ]</td>
<td>(SENSe):SWEep:MODE AUTO</td>
</tr>
<tr>
<td>[ MANUAL FREQ ]</td>
<td>(SENSe):FREQuency:MANual 0.015625 ~ 51200[HZ]</td>
</tr>
<tr>
<td>[ RESOLUTN SETUP ]</td>
<td>(SENSe):FREQuency:RESolution 0.015625 ~ 51200</td>
</tr>
<tr>
<td>[ RESOLUTN ]</td>
<td>[HZ</td>
</tr>
<tr>
<td>[ SWEEP LIN LOG ]</td>
<td>(SENSe):SWEep:SPACing LINear</td>
</tr>
<tr>
<td>[ AUTO RES ON OFF ]</td>
<td>(SENSe):FREQuency:RESolution:AUTO OFF</td>
</tr>
<tr>
<td>[ MAXIMUM % CHANGE ]</td>
<td>(SENSe):FREQuency:RESolution:AUTO:MCHange 0.391 ~ 100</td>
</tr>
<tr>
<td>[ MINIMUM RESOLUTN ]</td>
<td>(SENSe):FREQuency:RESolution:AUTO:MINimum 0.015625 ~ 51200</td>
</tr>
<tr>
<td>[ RESOLUTN ]</td>
<td>[HZ</td>
</tr>
</tbody>
</table>
## Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ SAMPLE TIME ]</td>
<td>[SENSe:SWEep:TIME 0.000976562 – 8192][S]</td>
</tr>
<tr>
<td>[ RECORD TIME ]</td>
<td>[SENSe:SWEep:TIME 0.000976562 – 8192][S]</td>
</tr>
<tr>
<td>[ BLOCKSIZE ]</td>
<td>[SENSe:FREQuency:BLOCkszie 256 – 2048]</td>
</tr>
<tr>
<td>[ HISTOGRAM LENGTH ]</td>
<td>[SENSe:AVERage:TIME 0 – 9.9e+37[S]</td>
</tr>
<tr>
<td>[ HISTOGRAM BINS ]</td>
<td>[SENSe:HISTogram:BINS 4 – 1024]</td>
</tr>
<tr>
<td>[ RECORD LENGTH ]</td>
<td>[SENSe:SWEep:TIME 0.000976562 – 8192][S]</td>
</tr>
</tbody>
</table>

### 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

<p>| [ CHANNEL 1 | 2 | 3 | 4 RANGE ] | [SENSe:VOLTage1 | 2 | 3 | 4[:OC]:RANGE[:UPPer] -51 – 31.66 |
| [ DBVRMS | VPK | DBVPK | V | DBV | EU | DEU | VRMS] |
| [ CH FIXED RANGE ] | [SENSe:VOLTage1 | 2 | 3 | 4[:OC]:RANGE:AUto OFF] |
| [ CH AUTO UP ONLY ] | [SENSe:VOLTage1 | 2 | 3 | 4[:OC]:RANGE:AUto ON;AUTO:DIRection UP] |
| [ CH AUTO RANGE ] | [SENSe:VOLTage1 | 2 | 3 | 4[:OC]:RANGE:AUto ON;AUTO:DIRection EITHer] |
| [ FRONT END CH1 | 2 | 3 | 4 SETUP ] | 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 |</p>
<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ XDCR UNIT CH1</td>
<td>2</td>
</tr>
<tr>
<td>[ XDCR UNIT ON OFF ]</td>
<td></td>
</tr>
<tr>
<td>[ XDCR UNIT SENSITIVITY ]</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>[ XDCR UNIT LABEL ]</td>
<td></td>
</tr>
<tr>
<td>[ Pascal ]</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( g )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( m/s^2 )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( m/s )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( m )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( kg )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( N )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( dyn )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( inch/s^2 )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( inch/s )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( inch )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( mil )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>( lb )</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>[ USER UNIT LABEL ]</td>
<td>[SENSe:VOLTage1</td>
</tr>
<tr>
<td>[ TACHOMETR SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[ TACH PULS PER REV ]</td>
<td>TRIGger:TACHometer:PCOunt 0.5 ~ 2048</td>
</tr>
<tr>
<td>[ TRG RANGE +/- 204 ]</td>
<td>TRIGger:TACHometer:RANGe HIGH</td>
</tr>
<tr>
<td>[ LEVEL ]</td>
<td>TRIGger:TACHometer:LEVel -20 ~ 20[V]</td>
</tr>
<tr>
<td>[ HOLDOFF TIME ]</td>
<td>TRIGger:TACHometer:HOLDoff 0 ~ 0.052224[S]</td>
</tr>
<tr>
<td>[ SLOPE POS NEG ]</td>
<td>TRIGger:TACHometer:SLOPe POSitive</td>
</tr>
<tr>
<td>[ TACH DISP ON OFF ]</td>
<td>DISPlay:RPM OFF</td>
</tr>
<tr>
<td>[ Source ]</td>
<td></td>
</tr>
<tr>
<td>[ SOURCE ON OFF ]</td>
<td>OUTPut[:STATe] OFF</td>
</tr>
<tr>
<td>[ LEVEL ]</td>
<td>SOURce:VOLTage[:LEVEL][:IMMediate][:AMPLitude]</td>
</tr>
<tr>
<td></td>
<td>{-9.9e+37 ~ 13.9794}[DBVRMS</td>
</tr>
<tr>
<td>[ DC OFFSET ]</td>
<td>SOURce:VOLTage[:LEVEL][:IMMediate][:AMPLitude]:OFFSET</td>
</tr>
<tr>
<td></td>
<td>{-9.9e+37 ~ 13.9794}[DBVRMS</td>
</tr>
<tr>
<td>[ RANDOM NOISE ]</td>
<td>SOURce:FUNCTION[:SHApe] RANDom</td>
</tr>
<tr>
<td>[ PERIODIC CHIRP ]</td>
<td>SOURce:FUNCTION[:SHApe] PCHirp</td>
</tr>
<tr>
<td>[ PINK NOISE ]</td>
<td>SOURce:FUNCTION[:SHApe] PINK</td>
</tr>
<tr>
<td>[ FIXED SINE ]</td>
<td>SOURce:FUNCTION[:SHApe] SIN;FREQuency[:CW] 0 ~ 115000[HZ]</td>
</tr>
<tr>
<td>[ BURST RANDOM ]</td>
<td>SOURce:FUNCTION[:SHApe] BRAN;BURSt 0 ~ 100[PCT]</td>
</tr>
<tr>
<td>[ BURST CHIRP ]</td>
<td>SOURce:FUNCTION[:SHApe] BRAN;BURSt 0 ~ 100[PCT]</td>
</tr>
<tr>
<td>[ ARBITRARY D1-D8 ]</td>
<td>SOURce:FUNCTION[:SHApe] USER [D1</td>
</tr>
<tr>
<td>[ CAPTURE CHANNEL ]</td>
<td>SOURce:USER:CAPTure 1 ~ 4</td>
</tr>
</tbody>
</table>
## Front Panel Key to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ARB SRC SETUP ]</td>
<td>SOURce:USER:REPeat OFF</td>
</tr>
<tr>
<td>[ REPEAT ON OFF ]</td>
<td>SOURce:USER:[REGister] D1</td>
</tr>
<tr>
<td>[ DATA REG D1 ]</td>
<td>SOURce:USER:[REGister] D2</td>
</tr>
<tr>
<td>[ ARB FILTER ON OFF ]</td>
<td>OUTPut:FILTER OFF</td>
</tr>
<tr>
<td>[ RAMP RATE ]</td>
<td>SOURce:VOLTage:SLEW 0 ~ 10000</td>
</tr>
<tr>
<td>[ AUTOLEVEL ON OFF ]</td>
<td>SOURce:VOLTage:[LEVel]:AUTO OFF</td>
</tr>
<tr>
<td>[ AUTOLEVEL SETUP ]</td>
<td>SOURce:VOLTage:[LEVel]:REFerence:CHANnel INPut1</td>
</tr>
<tr>
<td>[ REF CHAN CH1 CH2 CH3 CH4 ]</td>
<td>SOURce:VOLTage:[LEVel]:REFerence -69.276 ~ 31.66</td>
</tr>
<tr>
<td>[ REFERENCE LEVEL ]</td>
<td>SOURce:VOLTage:[LEVel]:REFerence -69.276 ~ 31.66</td>
</tr>
<tr>
<td>[ REFERENCE TOLERANCE ]</td>
<td>SOURce:VOLTage:[LEVel]:REFerence:TOLerance 0.1 ~ 20[DB]</td>
</tr>
<tr>
<td>[ FREE RUN TRIGGER ]</td>
<td>TRIGger:SOURce IMMEDIATE</td>
</tr>
<tr>
<td>[ EXTERNAL TRIGGER ]</td>
<td>TRIGger:SOURce EXTERNAL</td>
</tr>
<tr>
<td>[ CHANNEL 1</td>
<td>2</td>
</tr>
<tr>
<td>[ SOURCE TRIGGER ]</td>
<td>TRIGger:SOURce OUTPut</td>
</tr>
<tr>
<td>[ GPIB TRIGGER ]</td>
<td>TRIGger:SOURce BUS</td>
</tr>
<tr>
<td>[ TACHOMETER SETUP ]</td>
<td>TRIGger:TACHometer:PCOunt 0.5 ~ 2048</td>
</tr>
<tr>
<td>[ TACH PULS PER REV ]</td>
<td>TRIGger:TACHometer:RANge HIGH</td>
</tr>
<tr>
<td>[ TRG RANGE +/- 20 4 ]</td>
<td>TRIGger:TACHometer:LEVel -20 ~ 20[V]</td>
</tr>
<tr>
<td>[ LEVEL ]</td>
<td>TRIGger:TACHometer:HALF 0 ~ 0.052224[S]</td>
</tr>
<tr>
<td>[ HOLDOFF TIME ]</td>
<td>TRIGger:TACHometer:SLOPe POSitive</td>
</tr>
<tr>
<td>[ SLOPE POS NEG ]</td>
<td>TRIGger:TACHometer:LEVel -9.9e+37 ~ 9.9e+37</td>
</tr>
<tr>
<td>[ TRIGGER SETUP ]</td>
<td>TRIGger:LEVel -9.9e+37 ~ 9.9e+37</td>
</tr>
<tr>
<td>[ CHANNEL LEVEL ]</td>
<td>TRIGger:EXternal:RANge HIGH</td>
</tr>
<tr>
<td>[ EXT LEVEL USER ]</td>
<td>TRIGger:EXternal:RANge HIGH</td>
</tr>
<tr>
<td>[ EXT LEVEL TTL ]</td>
<td>TRIGger:EXternal:RANge HIGH</td>
</tr>
<tr>
<td>[ USER EXT LEVEL ]</td>
<td>TRIGger:EXternal:LEVel -10 ~ 10[V]</td>
</tr>
<tr>
<td>[ EXT RANGE +/- 10 2 ]</td>
<td>TRIGger:EXternal:RANge HIGH</td>
</tr>
<tr>
<td>[ SLOPE POS NEG ]</td>
<td>TRIGger:SLOPe POSitive</td>
</tr>
</tbody>
</table>
Cross-Reference from Front-Panel Keys to GPIB Commands

[ 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 INCREASING ] ARM:RPM:MODE UP
[ RPM DECREASING ] 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]

Start

START Hardkey ABORT::INITiate[:IMMEDIATE]

PauseCont

PAUSE/CONT Hardkey INITiate:CONTinuous OFF | 0 | ON | 1

Avg

[ AVERAGE ON OFF ] [SENSe:AVerage[:STATe] OFF | 0 | ON | 1
[ NUMBER AVERAGES ] [SENSe:AVerage:COUNT 1 ~ 1e+07
[ AVERAGE TYPE ] [SENSe:AVerage:TYPE RMS
[ RMS ] [SENSe:AVerage:TYPE RMS; TCONtrol EXPonential
[ RMS EXPONENTL ] [SENSe:AVerage:TYPE TIME
[ TIME ] [SENSe:AVerage:TYPE TIME; TCONtrol EXPonential
[ TIME EXPONENTL ] [SENSe:AVerage:TYPE MAXIMUM
[ PEAK HOLD ] [SENSe:AVerage:TYPE MAXIMUM
[ FAST AVG ON OFF ] [SENSe:AVerage:RESult[:STATe] OFF | 0 | ON | 1
[ UPDATE RATE ] [SENSe:AVerage:RESult:RATE 1 ~ 99999
[ REPEAT ON OFF ] [SENSe:AVerage:TCOntrl 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

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ AVERAGE PREVIEW ]</td>
<td>[SENSe:AVERage:PREView OFF ]</td>
</tr>
<tr>
<td>[ PREVIEW OFF ]</td>
<td>[SENSe:AVERage:PREView OFF ]</td>
</tr>
<tr>
<td>[ MANUAL PREVIEW ]</td>
<td>[SENSe:AVERage:PREView MANual ]</td>
</tr>
<tr>
<td>[ TIMED PREVIEW ]</td>
<td>[SENSe:AVERage:PREView TIMed ]</td>
</tr>
<tr>
<td>[ TIME ]</td>
<td>[SENSe:AVERage:PREView:TIME 0.1 ~ 3600[S] ]</td>
</tr>
<tr>
<td>[ REJECT TIME REC ]</td>
<td>[SENSe:AVERage:PREView:REJect ]</td>
</tr>
<tr>
<td>[ ACCEPT TIME REC ]</td>
<td>[SENSe:AVERage:PREView:ACcept ]</td>
</tr>
<tr>
<td>[ LINEAR ]</td>
<td>[SENSe:AVERage:TYPE RMS ]</td>
</tr>
<tr>
<td>[ EXPONENTL ]</td>
<td>[SENSe:AVERage:TYPE RMS;TCONtrol EXPonental ]</td>
</tr>
<tr>
<td>[ EQUAL CONFID ]</td>
<td>[SENSe:AVERage:TYPE ECONfidence ]</td>
</tr>
<tr>
<td>[ PEAK HOLD ]</td>
<td>[SENSe:AVERage:TYPE MAX ]</td>
</tr>
<tr>
<td>[ HOLD SETUP ]</td>
<td>[SENSe:AVERage:HOLD OFF ]</td>
</tr>
<tr>
<td>[ OFF ]</td>
<td>[SENSe:AVERage:HOLD MAXimum ]</td>
</tr>
<tr>
<td>[ MAXIMUM ]</td>
<td>[SENSe:AVERage:HOLD MINimum ]</td>
</tr>
<tr>
<td>[ MINIMUM ]</td>
<td>[SENSe:AVERage:TIME 0.125 ~ 8192[S] ]</td>
</tr>
<tr>
<td>[ AVERAGE TIME ]</td>
<td>[SENSe:AVERage:CONFidence 0 ~ 100[DB] ]</td>
</tr>
<tr>
<td>[ CONFIDNCE LEVEL ]</td>
<td>[SENSe:AVERage:IMPulse OFF</td>
</tr>
<tr>
<td>[ IMPULSE ON OFF ]</td>
<td>[SENSe:SWep:STIMe 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ SETTLE TIME ]</td>
<td>[SENSe:SWep:DWELi 0.00025 ~ 32768[S</td>
</tr>
<tr>
<td>[ INTEGRATE TIME ]</td>
<td></td>
</tr>
</tbody>
</table>
## 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 1,2| 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| 1,3| 1,4| 3,4';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

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ MORE CHOICES ]</td>
<td></td>
</tr>
<tr>
<td>[ COMP PWR CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XFR:POW:COMP 1</td>
</tr>
<tr>
<td>[ ORDER TRK CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ TRACK 1 ]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ TRACK 2 ]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ TRACK 3 ]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ TRACK 4 ]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ TRACK 5 ]</td>
<td>CALCulate#:FEED 'XORDer:TRACk 1</td>
</tr>
<tr>
<td>[ RPM PROFILE ]</td>
<td>CALCulate#:FEED 'XRPM:PROF';MATH:STATe OFF;*WAI</td>
</tr>
<tr>
<td>[ NORM VAR CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XFR:POW:VAR 1</td>
</tr>
<tr>
<td>[ AUTO CORR CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XTIM:CORR 1</td>
</tr>
<tr>
<td>[ CROS CORR 2/1 3/1/ 4/1 4/3 ]</td>
<td>CALCulate#:FEED 'XTIM:CORR:CROS 1,2</td>
</tr>
<tr>
<td>[ HISTOGRAM CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XTIM:VOLT:HIST 1</td>
</tr>
<tr>
<td>[ PDF CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XTIM:VOLT:PDF 1</td>
</tr>
<tr>
<td>[ CDF CHANNEL 1 2 3 4]</td>
<td>CALCulate#:FEED 'XTIM:VOLT:CDF 1</td>
</tr>
<tr>
<td>[ UNFILTERD TIME CH 1 2 3 4]</td>
<td>CALCulate#:FEED 'XTIM:VOLT 1</td>
</tr>
</tbody>
</table>

### Trace Coords

<p>| [ 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 APERATURE ]     | 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'                                                     |</p>
<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ORDER SETUP ]</td>
<td>CALCulate#:UNIT:X:ORDer:FACTor 1e-15~1e+15 [HZ/ORD</td>
</tr>
<tr>
<td>[ HZ/ORDER RATIO ]</td>
<td>CALCulate#:UNIT:X:ORDer:FACTor 1e-15~1e+15 [HZ/ORD</td>
</tr>
<tr>
<td>[ TRACE RPM ]</td>
<td>CALCulate#:UNIT:X:ORDer:FACTor 1e-15~1e+15 [HZ/ORD</td>
</tr>
<tr>
<td>[ USER X SETUP ]</td>
<td>CALCulate#:UNIT:X:USER:FREQuency:FACTor 1e-15~1e+15</td>
</tr>
<tr>
<td>[ USER FREQ FACTOR ]</td>
<td>CALCulate#:UNIT:X:USER:FREQuency:FACTor 1e-15~1e+15</td>
</tr>
<tr>
<td>[ USER TIME FACTOR ]</td>
<td>CALCulate#:UNIT:X:USER:FREQuency:LABel '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ FREQ LABEL ]</td>
<td>CALCulate#:UNIT:X:USER:FREQuency:LABel '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ TIME LABEL ]</td>
<td>CALCulate#:UNIT:X:USER:FREQuency:LABel '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ Y UNITS]</td>
<td>CALCulate#:UNIT:AMPLitude PEAK</td>
</tr>
<tr>
<td>[ AMPLITUDE PK PP RMS ]</td>
<td>CALCulate#:UNIT:AMPLitude PEAK</td>
</tr>
<tr>
<td>[ PHASE DEG RAD ]</td>
<td>CALCulate#:UNIT:AMPLitude PEAK</td>
</tr>
<tr>
<td>[ dB REF SETUP ]</td>
<td>CALCulate#:UNIT:DBReference 'DBV'</td>
</tr>
<tr>
<td>[ dBV dBUEU ]</td>
<td>CALCulate#:UNIT:DBReference 'DBV'</td>
</tr>
<tr>
<td>[ dBm ]</td>
<td>CALCulate#:UNIT:DBReference 'DBM'</td>
</tr>
<tr>
<td>[ dBSPL 20uPa ]</td>
<td>CALCulate#:UNIT:DBReference 'DBSPL'</td>
</tr>
<tr>
<td>[ USER REFERENCE ]</td>
<td>CALCulate#:UNIT:DBReference 'USER'</td>
</tr>
<tr>
<td>[ dBm REF IMPEDANCE]</td>
<td>CALCulate#:UNIT:DBReference:IMPedance 1e-15~1e+15[OHM]</td>
</tr>
<tr>
<td>[ USER REF LEVEL ]</td>
<td>CALCulate#:UNIT:DBReference REFerence 1e-15~1e+15</td>
</tr>
<tr>
<td>[ USER LABEL ]</td>
<td>CALCulate#:UNIT:DBReference REFerence:LABel '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ XDCR UNIT CONVERT ]</td>
<td>CALCulate#:UNIT:MECHanical 'G'</td>
</tr>
<tr>
<td>[ g (ACCEL) ]</td>
<td>CALCulate#:UNIT:MECHanical 'G'</td>
</tr>
<tr>
<td>[ m/s^2 (ACCEL) ]</td>
<td>CALCulate#:UNIT:MECHanical 'M/S2'</td>
</tr>
<tr>
<td>[ m/s (VEL) ]</td>
<td>CALCulate#:UNIT:MECHanical 'M/S'</td>
</tr>
<tr>
<td>[ m (DISP) ]</td>
<td>CALCulate#:UNIT:MECHanical 'M'</td>
</tr>
<tr>
<td>[ inch/s^2 (ACCEL) ]</td>
<td>CALCulate#:UNIT:MECHanical 'INCH/S2'</td>
</tr>
<tr>
<td>[ inch/s (VEL) ]</td>
<td>CALCulate#:UNIT:MECHanical 'INCH/S'</td>
</tr>
<tr>
<td>[ inch (DISP) ]</td>
<td>CALCulate#:UNIT:MECHanical 'INCH'</td>
</tr>
<tr>
<td>[ mil (DISP) ]</td>
<td>CALCulate#:UNIT:MECHanical 'MIL'</td>
</tr>
<tr>
<td>[ VOLTS ]</td>
<td>CALCulate#:UNIT:VOLT 'V'</td>
</tr>
<tr>
<td>[ VOLTS^2 ]</td>
<td>CALCulate#:UNIT:VOLT 'V2'</td>
</tr>
<tr>
<td>[ V/Hz ]</td>
<td>CALCulate#:UNIT:VOLT 'V/Hz'</td>
</tr>
<tr>
<td>[ V^2/Hz PSD ]</td>
<td>CALCulate#:UNIT:VOLT 'V2/Hz'</td>
</tr>
<tr>
<td>[ V^2s/Hz ESD ]</td>
<td>CALCulate#:UNIT:VOLT 'V2s/Hz'</td>
</tr>
<tr>
<td>[ X-AXIS LIN LOG ]</td>
<td>DISPlay:WINDow#:TRACe:X:SPACing LiNear</td>
</tr>
</tbody>
</table>
## Front Panel Key to GPIB Commands

<table>
<thead>
<tr>
<th>Scale</th>
</tr>
</thead>
</table>

| [ AUTOSCALE ON OFF ] | DISPlay:WINdow#:TRACe:Y[:SCALe]AUTO | OFF | 0 | ON | 1 | ONCE |
| [ TOP REFERENCE ] | DISPlay:WINdow#:TRACe:Y[:SCALe]:REfer:TOP | (-9.9e+37 ~ 9.9e+37 | DBVRMS | VRMS | VPK | DBVPK | V | DBV | EU | DBEU) |
| [ CENTER REFERENCE ] | DISPlay:WINdow#:TRACe:Y[:SCALe]:REfer:CENTer | (-9.9e+37 ~ 9.9e+37 | DBVRMS | VRMS | VPK | DBVPK | V | DBV | EU | DBEU) |
| [ BOTTOM REFERENCE ] | DISPlay:WINdow#:TRACe:Y[:SCALe]:REfer:BOTtom | (-9.9e+37 ~ 9.9e+37 | DBVRMS | VRMS | VPK | DBVPK | V | DBV | EU | DBEU) |
| [ INP RANGE TRACKING ] | DISPlay:WINdow#:TRACe:Y[:SCALe]:REfer: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 |

<table>
<thead>
<tr>
<th>Active Trace</th>
</tr>
</thead>
</table>

<p>| [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 |</p>
<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[DEFINE FUNCTION]</td>
<td>CALCulate#:MATH:EXPRession1</td>
</tr>
<tr>
<td>[DEFINE F1]</td>
<td></td>
</tr>
<tr>
<td>[MEAS DATA]</td>
<td></td>
</tr>
<tr>
<td>[PWR SPEC CHANNEL 1 2 3 4]</td>
<td>PSPEC1</td>
</tr>
<tr>
<td>[+]</td>
<td>+</td>
</tr>
<tr>
<td>[-]</td>
<td>-</td>
</tr>
<tr>
<td>[*]</td>
<td>*</td>
</tr>
<tr>
<td>[/]</td>
<td>/</td>
</tr>
<tr>
<td>[LIN SPEC CHANNEL 1 2 3 4]</td>
<td>LSPEC1</td>
</tr>
<tr>
<td>[FREQ RESP]</td>
<td>FRES21</td>
</tr>
<tr>
<td>[COHERENCE]</td>
<td>COH21</td>
</tr>
<tr>
<td>[CROS SPEC]</td>
<td>CSPEC21</td>
</tr>
<tr>
<td>[TIME CHANNEL 1 2 3 4]</td>
<td>TIME1</td>
</tr>
<tr>
<td>[WINDO TIME CHANNEL 1 2 3 4]</td>
<td>WTIME1</td>
</tr>
<tr>
<td>[RESAMPLED TIME CH 1 2 3 4]</td>
<td>TIME1</td>
</tr>
<tr>
<td>[COMP PWR CHANNEL 1 2 3 4]</td>
<td>CPOW1</td>
</tr>
<tr>
<td>[ORDER TRK CHANNEL 1 2 3 4]</td>
<td>TRACK1</td>
</tr>
<tr>
<td>[RPM PROFILE]</td>
<td>RPM</td>
</tr>
<tr>
<td>[AUTO CORR CHANNEL 1 2 3 4]</td>
<td>ACORR1</td>
</tr>
<tr>
<td>[CROS CORR]</td>
<td>XCORR21</td>
</tr>
<tr>
<td>[HISTOGRAM CHANNEL 1 2 3 4]</td>
<td>HIST1</td>
</tr>
<tr>
<td>[PDF CHANNEL 1 2 3 4]</td>
<td>PDF1</td>
</tr>
<tr>
<td>[CDF CHANNEL 1 2 3 4]</td>
<td>CDF1</td>
</tr>
<tr>
<td>[DATA REG D1-D8]</td>
<td>D1</td>
</tr>
<tr>
<td>[DATA REG D1]</td>
<td>D1</td>
</tr>
<tr>
<td>[DATA REG D2]</td>
<td>D2</td>
</tr>
<tr>
<td>[DATA REG D3]</td>
<td>D3</td>
</tr>
<tr>
<td>[DATA REG D4]</td>
<td>D4</td>
</tr>
<tr>
<td>[DATA REG D5]</td>
<td>D5</td>
</tr>
<tr>
<td>[DATA REG D6]</td>
<td>D6</td>
</tr>
<tr>
<td>[DATA REG D7]</td>
<td>D7</td>
</tr>
<tr>
<td>[DATA REG D8]</td>
<td>D8</td>
</tr>
<tr>
<td>[CONSTANT K1-K5]</td>
<td>K1</td>
</tr>
<tr>
<td>[CONSTANT K1]</td>
<td>K1</td>
</tr>
<tr>
<td>[CONSTANT K2]</td>
<td>K2</td>
</tr>
<tr>
<td>[CONSTANT K3]</td>
<td>K3</td>
</tr>
<tr>
<td>[CONSTANT K4]</td>
<td>K4</td>
</tr>
<tr>
<td>[CONSTANT K5]</td>
<td>K5</td>
</tr>
<tr>
<td>Front Panel Key</td>
<td>GPIB Command</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>[ FUNCTION F1-F5 ]</td>
<td>F1 - F5</td>
</tr>
<tr>
<td>[ FUNCTION F1 ]</td>
<td>F1</td>
</tr>
<tr>
<td>[ FUNCTION F2 ]</td>
<td>F2</td>
</tr>
<tr>
<td>[ FUNCTION F3 ]</td>
<td>F3</td>
</tr>
<tr>
<td>[ FUNCTION F4 ]</td>
<td>F4</td>
</tr>
<tr>
<td>[ FUNCTION F5 ]</td>
<td>F5</td>
</tr>
<tr>
<td>[ OPERATION ]</td>
<td>CONJ(</td>
</tr>
<tr>
<td></td>
<td>MAG(</td>
</tr>
<tr>
<td></td>
<td>REAL(</td>
</tr>
<tr>
<td></td>
<td>IMAG(</td>
</tr>
<tr>
<td></td>
<td>SQRT(</td>
</tr>
<tr>
<td></td>
<td>FFT(</td>
</tr>
<tr>
<td></td>
<td>INVERSE FFT(</td>
</tr>
<tr>
<td></td>
<td>PSD(</td>
</tr>
<tr>
<td>[ MORE ]</td>
<td>LN(</td>
</tr>
<tr>
<td></td>
<td>EXP(</td>
</tr>
<tr>
<td></td>
<td>'*JOMEGA(</td>
</tr>
<tr>
<td></td>
<td>'-JOMEGA(</td>
</tr>
<tr>
<td></td>
<td>AWEIGHT(</td>
</tr>
<tr>
<td></td>
<td>BWEIGHT(</td>
</tr>
<tr>
<td></td>
<td>CWEIGHT(</td>
</tr>
<tr>
<td></td>
<td>DIFF(</td>
</tr>
<tr>
<td></td>
<td>INTEG(</td>
</tr>
<tr>
<td></td>
<td>DEFINE F2</td>
</tr>
<tr>
<td></td>
<td>DEFINE F3</td>
</tr>
<tr>
<td></td>
<td>DEFINE F4</td>
</tr>
<tr>
<td></td>
<td>DEFINE F5</td>
</tr>
<tr>
<td>[ DEFINE CONSTANT ]</td>
<td>CALCulate:MATH:CONStant1 - 9.9e+37 - 9.9e+37</td>
</tr>
<tr>
<td></td>
<td>DEFINE K2</td>
</tr>
<tr>
<td></td>
<td>DEFINE K3</td>
</tr>
<tr>
<td></td>
<td>DEFINE K4</td>
</tr>
<tr>
<td></td>
<td>DEFINE K5</td>
</tr>
<tr>
<td>[ LIMIT TEST ]</td>
<td>DISPLAY:WINDow:LIMIT:STATe OFF</td>
</tr>
<tr>
<td></td>
<td>TEST EVAL ON OFF</td>
</tr>
<tr>
<td></td>
<td>FAIL BEEP ON OFF</td>
</tr>
<tr>
<td>[ DEFINE UPPER LIM ]</td>
<td>CALCulate#:LIMIT:UPPer:SEGMENT -9.9e+37 - 9.9e+37</td>
</tr>
<tr>
<td>[ START SEGMENT ]</td>
<td>CALCulate#:LIMIT:UPPer:SEGMENT:CLEar</td>
</tr>
<tr>
<td>[ MOVE ALL VERTICAL ]</td>
<td>CALCulate#:LIMIT:UPPer:SEGMENT:CLEar</td>
</tr>
<tr>
<td>[ DELETE SEGMENT ]</td>
<td>CALCulate#:LIMIT:UPPer:SEGMENT:CLEar</td>
</tr>
<tr>
<td>Front Panel Key</td>
<td>GPIB Command</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>[ DELETE ALL ]</td>
<td>CALCulate #:LIMIT:UPPer:CLEar</td>
</tr>
<tr>
<td>[ CONFIRM/DELETE ]</td>
<td>CALCulate #:LIMIT:UPPer:TRACe[:IMMediate]</td>
</tr>
<tr>
<td>[ TRACE TO LIMIT ]</td>
<td>CALCulate #:LIMIT:UPPer:CLEar</td>
</tr>
<tr>
<td>[ DEFINE LOWER LIM ]</td>
<td>CALCulate #:LIMIT:UPPer:TRACe[:IMMediate]</td>
</tr>
<tr>
<td>[ START SEGMENT ]</td>
<td>CALCulate #:LIMIT:UPPer:TRACe[:IMMediate]</td>
</tr>
<tr>
<td>[ DELETE SEGMENT ]</td>
<td>CALCulate #:LIMIT:LOWer:SEGMENT:CLEar</td>
</tr>
<tr>
<td>[ DELETE ALL ]</td>
<td>CALCulate #:LIMIT:LOWer:CLEar</td>
</tr>
<tr>
<td>[ TRACE TO LIMIT ]</td>
<td>CALCulate #:LIMIT:LOWer:TRACe[:IMMediate]</td>
</tr>
<tr>
<td>[ DATA TABLE ]</td>
<td>CALCulate #:LIMIT:LOWer:CLEar</td>
</tr>
<tr>
<td>[ TABLE ON OFF ]</td>
<td>Display:WINDOW#:DTABle[:STATE] OFF</td>
</tr>
<tr>
<td>[ MARKER ON OFF ]</td>
<td>Display:WINDOW#:DTABle:MARKer[:STATE] OFF</td>
</tr>
<tr>
<td>[ SELECT X ENTRY ]</td>
<td>CALCulate #:MARKer:DTABle:X:SELECT[:POINT] 1 ~ 50</td>
</tr>
<tr>
<td>[ INSERT X VALUE ]</td>
<td>CALCulate #:MARKer:DTABle:X:INSert -9.9e+37 ~ 9.9e+37</td>
</tr>
<tr>
<td>[ ENTRY LABEL ]</td>
<td>CALCulate #:MARKer:DTABle:X:LABel '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ DELETE X ]</td>
<td>CALCulate #:MARKer:DTABle:X:DElete</td>
</tr>
<tr>
<td>[ CLEAR TABLE ]</td>
<td>CALCulate #:MARKer:DTABle:CLEar[:IMMediate]</td>
</tr>
<tr>
<td>[ COPY TABLE ]</td>
<td>CALCulate #:MARKer:DTABle:COPY[1</td>
</tr>
<tr>
<td>[ Y VALUES ABS REL ]</td>
<td>CALCulate #:MARKer:MODE ABSolute</td>
</tr>
<tr>
<td>[ ABORT FIT ]</td>
<td>CALCulate2:CFIT:ABORt</td>
</tr>
<tr>
<td>[ CURVE FIT ]</td>
<td>CALCulate2:CFIT[:IMMediate]</td>
</tr>
<tr>
<td>[ START FIT ]</td>
<td>CALCulate2:CFIT:ABORt</td>
</tr>
<tr>
<td>[ CURVE FIT REGISTER ]</td>
<td>CALCulate #:FIT:DESTination D1</td>
</tr>
<tr>
<td>[ D1 ]</td>
<td>CALCulate #:FIT:DESTination D2</td>
</tr>
<tr>
<td>[ D2 ]</td>
<td>CALCulate #:FIT:DESTination D3</td>
</tr>
<tr>
<td>[ D3 ]</td>
<td>CALCulate #:FIT:DESTination D4</td>
</tr>
<tr>
<td>[ D4 ]</td>
<td>CALCulate #:FIT:DESTination D5</td>
</tr>
<tr>
<td>[ D5 ]</td>
<td>CALCulate #:FIT:DESTination D6</td>
</tr>
<tr>
<td>[ D6 ]</td>
<td>CALCulate #:FIT:DESTination D7</td>
</tr>
<tr>
<td>[ D7 ]</td>
<td>CALCulate #:FIT:DESTination D8</td>
</tr>
<tr>
<td>[ D8 ]</td>
<td>CALCulate #:FIT:DESTination Dn</td>
</tr>
<tr>
<td>[ COPY FROM SYNTHESIS ]</td>
<td>CALCulate #:FIT:COPY SYThesis</td>
</tr>
<tr>
<td>[ FIT REGION ]</td>
<td>CALCulate #:FIT:FREQuency[:AUTO] ON</td>
</tr>
<tr>
<td>[ FULL SPAN ]</td>
<td>CALCulate #:FIT:FREQuency[:AUTO] OFF</td>
</tr>
<tr>
<td>[ USER SPAN ]</td>
<td>CALCulate #:FIT:FREQuency:STARt 0 ~ 115000(HZ)</td>
</tr>
<tr>
<td>[ START ]</td>
<td>CALCulate #:FIT:FREQuency:STOP 0.390625 ~ 115000(HZ)</td>
</tr>
</tbody>
</table>
| [ STOP ] | }
Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURVE FIT SETUP</td>
<td></td>
</tr>
<tr>
<td>ORDER MAX FIXED</td>
<td>CALCulate:CFIT:ORDer:AUTO OFF</td>
</tr>
<tr>
<td>NUMBER OF POLES</td>
<td>CALCulate:CFIT:ORDer:POLes 0 ~ 20</td>
</tr>
<tr>
<td>NUMBER OF ZEROS</td>
<td>CALCulate:CFIT:ORDer:ZERos 0 ~ 20</td>
</tr>
<tr>
<td>WEIGHT AUTO USER</td>
<td>CALCulate:CFIT:WEIGht:AUTO OFF</td>
</tr>
<tr>
<td>WEIGHT REGISTER</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>CALCulate :CFIT:WEIGht:REGister D1</td>
</tr>
<tr>
<td>D2</td>
<td>CALCulate:CFIT:WEIGht:REGister D2</td>
</tr>
<tr>
<td>D3</td>
<td>CALCulate:CFIT:WEIGht:REGister D3</td>
</tr>
<tr>
<td>D4</td>
<td>CALCulate:CFIT:WEIGht:REGister D4</td>
</tr>
<tr>
<td>D5</td>
<td>CALCulate:CFIT:WEIGht:REGister D5</td>
</tr>
<tr>
<td>D6</td>
<td>CALCulate:CFIT:WEIGht:REGister D6</td>
</tr>
<tr>
<td>D7</td>
<td>CALCulate:CFIT:WEIGht:REGister D7</td>
</tr>
<tr>
<td>D8</td>
<td>CALCulate:CFIT:WEIGht:REGister D8</td>
</tr>
<tr>
<td>TIME DELAY</td>
<td>CALCulate:CFIT:TDELay -100 ~ 100[S]</td>
</tr>
<tr>
<td>FREQUENCY SCALE</td>
<td>CALCulate:CFIT:FSCale 1e-06 ~ 1000000</td>
</tr>
<tr>
<td>TABLE ON OFF</td>
<td>DISPlay:VIEW CTAB</td>
</tr>
<tr>
<td>SYNTHESIS</td>
<td></td>
</tr>
<tr>
<td>START SYNTHESIS</td>
<td>CALCulate:SYNThesis[:IMMediate]</td>
</tr>
<tr>
<td>SYNTHESIS REGISTER</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>CALCulate#:SYNThesis:DESTination D1</td>
</tr>
<tr>
<td>D2</td>
<td>CALCulate:SYNThesis:DESTination D2</td>
</tr>
<tr>
<td>D3</td>
<td>CALCulate:SYNThesis:DESTination D3</td>
</tr>
<tr>
<td>D4</td>
<td>CALCulate:SYNThesis:DESTination D4</td>
</tr>
<tr>
<td>D5</td>
<td>CALCulate:SYNThesis:DESTination D5</td>
</tr>
<tr>
<td>D6</td>
<td>CALCulate:SYNThesis:DESTination D6</td>
</tr>
<tr>
<td>D7</td>
<td>CALCulate:SYNThesis:DESTination D7</td>
</tr>
<tr>
<td>D8</td>
<td>CALCulate:SYNThesis:DESTination D8</td>
</tr>
<tr>
<td>COPY FROM CURVE FIT</td>
<td>CALCulate:SYNThesis:COPY CFIT</td>
</tr>
<tr>
<td>CONVERT TABLE</td>
<td></td>
</tr>
<tr>
<td>CONVRT TO POLE ZERO</td>
<td>CALCulate :SYNThesis:TTYP# PZERO</td>
</tr>
<tr>
<td>CONVRT TO POLE RESD</td>
<td>CALCulate:SYNThesis:TTYPE PFRaction</td>
</tr>
<tr>
<td>CONVRT TO POLYNMIAL</td>
<td>CALCulate:SYNThesis:TTYPE POLynomial</td>
</tr>
<tr>
<td>SYNTHESIS SETUP</td>
<td></td>
</tr>
<tr>
<td>GAIN FACTOR</td>
<td>CALCulate2:SYNThesis:GAIN -9.9e37 ~ 9.9e37</td>
</tr>
<tr>
<td>TIME DELAY</td>
<td>CALCulate2:SYNThesis:TDELay -100 ~ 100[S]</td>
</tr>
<tr>
<td>FREQUENCY SCALE</td>
<td>CALCulate2:SYNThesis:FSCale 1e-06 ~ 1000000</td>
</tr>
<tr>
<td>X-AXIS LIN LOG</td>
<td>CALCulate2:SYNThesis:SPACing LNear</td>
</tr>
<tr>
<td>TABLE ON OFF</td>
<td>DISPlay:VIEW STAB</td>
</tr>
<tr>
<td>Front Panel Key</td>
<td>GPIB Command</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>DispFormat</td>
<td></td>
</tr>
<tr>
<td>[ SINGLE ]</td>
<td>DISPlay:FORMat SINGLE</td>
</tr>
<tr>
<td>[ SINGLE FRNT/ BACK ]</td>
<td>DISPlay:FORMat FBACk</td>
</tr>
<tr>
<td>[ UPPER/ LOWER ]</td>
<td>DISPlay:FORMat ULOWer</td>
</tr>
<tr>
<td>[ UPPER/BIG LOWER ]</td>
<td>DISPlay:FORMat SUBL</td>
</tr>
<tr>
<td>[ UPPR/LOWR FRNT/ BACK ]</td>
<td>DISPlay:FORMat ULFB</td>
</tr>
<tr>
<td>[ QUAD ]</td>
<td>DISPlay:FORMat QUAD</td>
</tr>
<tr>
<td>[ MEASURMT STATE ]</td>
<td>DISPlay:VIEW MS tate</td>
</tr>
<tr>
<td>[ WATERFALL SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[ WATERFALL ON OFF ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:STATe OFF</td>
</tr>
<tr>
<td>[ Z-AXIS TOP ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:TOP 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ Z-AXIS BOTTOM ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:BOTTOM 0 ~ 9.9e+37[S</td>
</tr>
<tr>
<td>[ Z AXIS RANGE ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:COUNT 1 ~ 50000</td>
</tr>
<tr>
<td>[ SKEW ON OFF ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:SKEW OFF</td>
</tr>
<tr>
<td>[ SKEW ANGLE ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:ANGLES 0 ~ 45</td>
</tr>
<tr>
<td>[ BASELINE SUPPRESS ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:BASEline 0 ~ 100[PCT]</td>
</tr>
<tr>
<td>[ WATERFALL STEPS ]</td>
<td>CALCulate#:WATERfall:COUNT 1 ~ 32767</td>
</tr>
<tr>
<td>[ MORE ]</td>
<td></td>
</tr>
<tr>
<td>[ TRACE HEIGHT ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:HEIGHT 1 ~ 100[PCT]</td>
</tr>
<tr>
<td>[ HIDN LINE REMOVED ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:HIDDEN OFF</td>
</tr>
<tr>
<td>[ HIDN LINE DISPLAYED ]</td>
<td>DISPlay[:WINDOW#:WATERfall]:HIDDEN OFF</td>
</tr>
<tr>
<td>[ GRID ON OFF ]</td>
<td>DISPlay[:WINDow]:TRACe:GRID[:STATe] OFF</td>
</tr>
<tr>
<td>[ BODE DIAGRAM ]</td>
<td>DISPlay:BODE</td>
</tr>
<tr>
<td>[ TACH DISP ON OFF ]</td>
<td>DISPlay:RPM[:STATe] OFF</td>
</tr>
<tr>
<td>[ TRACE TITLE ]</td>
<td>DISPlay[:WINDow]:TRACe:LAbel ‘&lt;STRING&gt;’</td>
</tr>
<tr>
<td>[ DFLT TITL ON OFF ]</td>
<td>DISPlay[:WINDow]:TRACe:LAbel:DEfault[:STATe] OFF</td>
</tr>
<tr>
<td>[ OCTAVE BND SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[ OVERALL ON OFF ]</td>
<td>DISPlay[:WINDow]:TRACe:BPoWer[:STATe] OFF</td>
</tr>
<tr>
<td>[ WEIGHTED ON OFF ]</td>
<td>DISPlay[:WINDow]:TRACe:APoWer[:STATe] OFF</td>
</tr>
<tr>
<td>[ MORE ]</td>
<td></td>
</tr>
<tr>
<td>[ ANNOTATN ON OFF ]</td>
<td>DISPlay:ANNotation[:ALL] OFF</td>
</tr>
<tr>
<td>[ DISPLAY ON OFF ]</td>
<td>DISPlay:ENABle OFF</td>
</tr>
<tr>
<td>[ DISPLAY BRIGHTNESS ]</td>
<td>DISPlay:BRIGhtness 0.5 ~ 1</td>
</tr>
<tr>
<td>[ EXT MON ON OFF ]</td>
<td>DISPlay:EXTernal[:STATe] OFF</td>
</tr>
</tbody>
</table>
### Cross-Reference from Front-Panel Keys to GPIB Commands

#### Marker Group

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ MARKER ON OFF ]</td>
<td>CALCulate#:MARKer:STATe OFF</td>
</tr>
<tr>
<td>[ COUPLED ON OFF ]</td>
<td>CALCulate#:MARKer:COUPled:STATe OFF</td>
</tr>
<tr>
<td>[ MARKER X ENTRY ]</td>
<td>CALCulate#:MARKer:X:ABSolute -9.9e+37 ~ 9.9e+37 [HZ</td>
</tr>
<tr>
<td>[ MKR VALUE ABS REL ]</td>
<td>CALCulate#:MARKer:MODE ABSolute</td>
</tr>
<tr>
<td>[ REFERENCE TO MARKER ]</td>
<td>CALCulate#:MARKer:MODE RELative;REFerence:X (CALC#:MARK:X? ;Y CALC#:MARK:Y?)</td>
</tr>
<tr>
<td>[ REFERENCE SETUP ]</td>
<td>CALCulate#:MARKer:REFerence:MODE RELative;X</td>
</tr>
<tr>
<td>[ REFERENCE X ENTRY ]</td>
<td>CALCulate#:MARKer:REFerence:X -9.9e+37 ~ 9.9e+37 [HZ</td>
</tr>
<tr>
<td>[ REFERENCE Y ENTRY ]</td>
<td>CALCulate#:MARKer:REFerence:Y -9.9e+37 ~ 9.9e+37 [DBVRMS</td>
</tr>
<tr>
<td>[ PEAK TRK ON OFF ]</td>
<td>CALCulate#:MARKer:MAXimum:[GLOBal]TRACk OFF</td>
</tr>
<tr>
<td>[ NEXT PEAK RIGHT ]</td>
<td>CALCulate#:MARKer:MAXimum:RIGHt</td>
</tr>
<tr>
<td>[ NEXT PEAK LEFT ]</td>
<td>CALCulate#:MARKer:MAXimum:LEFT</td>
</tr>
<tr>
<td>[ MARKER TO PEAK ]</td>
<td>CALCulate#:MARKer:MAXimum:[GLOBal]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marker Fctn</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ MARKER FCTN OFF ]</td>
</tr>
<tr>
<td>[ HARMONIC MARKER ]</td>
</tr>
<tr>
<td>[ NUMBER OF HARMONICS ]</td>
</tr>
<tr>
<td>[ COMPUTE OFF ]</td>
</tr>
<tr>
<td>[ THD ]</td>
</tr>
<tr>
<td>[ HARMONIC POWER ]</td>
</tr>
<tr>
<td>[ MARKRS TO DATA TABL ]</td>
</tr>
<tr>
<td>[ DATA TABL ON OFF ]</td>
</tr>
<tr>
<td>[ BAND MARKER ]</td>
</tr>
<tr>
<td>[ BAND CENTER ]</td>
</tr>
<tr>
<td>[BAND START ]</td>
</tr>
<tr>
<td>[ BAND STOP ]</td>
</tr>
</tbody>
</table>
## Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ COMPUTE OFF ]</td>
<td>CALCulate#:MARKer:FUNCtion OFF</td>
</tr>
<tr>
<td>[ BAND POWER ]</td>
<td>CALCulate#:MARKer:FUNCtion BPOWer</td>
</tr>
<tr>
<td>[ RMS SQRT(PWR) ]</td>
<td>CALCulate#:MARKer:FUNCtion BRMS</td>
</tr>
<tr>
<td>[ SPECTRUM ]</td>
<td>CALCulate#:MARKer:FUNCtion SGMS</td>
</tr>
<tr>
<td>[ SPECTRUM MARKER ]</td>
<td>CALCulate#:MARKer:MARKer:FUNCtion SPMK</td>
</tr>
<tr>
<td>[ SPECTRUM INCREMENT ]</td>
<td>CALCulate#:MARKer:MARKer:FUNCtion SINC</td>
</tr>
<tr>
<td>[ NUMBER OF SPECTRUM BANDS ]</td>
<td>CALCulate#:MARKer:MARKer:COUNT 0~200</td>
</tr>
<tr>
<td>[ COMPUTE OFF ]</td>
<td>CALCulate#:MARKer:FUNCtion OFF</td>
</tr>
<tr>
<td>[ MARKERS TO DATA TABL ]</td>
<td>CALCulate#:MARKer:DTABle:X:SELect</td>
</tr>
<tr>
<td>[ DATA TABL ON OFF ]</td>
<td>DISPlay:WINDow#:DTABle[:STATe]</td>
</tr>
<tr>
<td>[ WATERFALL MARKERS ]</td>
<td>DISPlay:WINDow#:WATerfall[:STATe]</td>
</tr>
<tr>
<td>[ TRACE SELECT ]</td>
<td>CALC#:WATerfall:TRACe:SELect -9.9e+37~9.9e+37</td>
</tr>
<tr>
<td>[ SLICE SELECT ]</td>
<td>CALC#:WATerfall:SLICe:SELect -9.9e+37~9.9e+37</td>
</tr>
<tr>
<td>[ SAVE AND DISP DATA ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ SAVE TO DATA REG ]</td>
<td></td>
</tr>
<tr>
<td>[ D1 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D2 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D3 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D4 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D5 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D6 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D7 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ D8 ]</td>
<td>CALC#:WATerfall:TRACe:SELect;</td>
</tr>
<tr>
<td>[ WATERFALL SETUP ]</td>
<td>DISPlay:WINDow#:WATerfall[:STATe]</td>
</tr>
<tr>
<td>[ Z-AXIS TOP ]</td>
<td>DISPlay:WINDow#:WATerfall:TOP 0~9.9e+37</td>
</tr>
<tr>
<td>[ Z-AXIS BOTTOM ]</td>
<td>DISPlay:WINDow#:WATerfall:BOTTOM 0~9.9e+37</td>
</tr>
<tr>
<td>[ Z AXIS RANGE ]</td>
<td>DISPlay:WINDow#:WATerfall:COUNT 1~50000</td>
</tr>
<tr>
<td>[ SKEW ON OFF ]</td>
<td>DISPlay:WINDow#:WATerfall:SKEW OFF</td>
</tr>
<tr>
<td>[ SKEW ANGLE ]</td>
<td>DISPlay:WINDow#:WATerfall:SKEW:ANGLe 0~45</td>
</tr>
<tr>
<td>[ BASELINE SUPPRESS ]</td>
<td>DISPlay:WINDow#:WATerfall:BASEline 0~100[PCT]</td>
</tr>
<tr>
<td>[ WATERFALL STEPS ]</td>
<td>CALCulate#:WATerfall:COUNT 1~32767</td>
</tr>
<tr>
<td>[ MORE ]</td>
<td>DISPlay:WINDow#:WATerfall:HEIght 1~100[PCT]</td>
</tr>
<tr>
<td>[ HIDN LINE REMOVED ]</td>
<td>DISPlay:WINDow#:WATerfall:HIDDen ON</td>
</tr>
<tr>
<td>[ HIDN LINE DISPLAYED ]</td>
<td>DISPlay:WINDow#:WATerfall:HIDDen OFF</td>
</tr>
</tbody>
</table>
Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[ TIME PARAMETERS]</strong></td>
<td></td>
</tr>
<tr>
<td>[START TIME]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STARt -9.9e+37 \sim 9.9e+37[S</td>
</tr>
<tr>
<td>[STOP TIME]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STOP -9.9e+37 \sim 9.9e+37[S</td>
</tr>
<tr>
<td>[COMPUTE OFF]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion OFF}</td>
</tr>
<tr>
<td>[OVERSHOOT]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion OVERshoot}</td>
</tr>
<tr>
<td>[RISE TIME]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion RTIME}</td>
</tr>
<tr>
<td>[SETTLING TIME]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion STIME}</td>
</tr>
<tr>
<td>[DELAY TIME]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion DTIME}</td>
</tr>
<tr>
<td><strong>[ GAIN/PHAS MARGINS]</strong></td>
<td></td>
</tr>
<tr>
<td>[START FREQUENCY]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STARt 0 \sim 102400(HZ)}</td>
</tr>
<tr>
<td>[STOP FREQUENCY]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STOP 0 \sim 102400(HZ)}</td>
</tr>
<tr>
<td>[COMPUTE OFF]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion OFF}</td>
</tr>
<tr>
<td>[COMPUTE MARGINS]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion GMARgin}</td>
</tr>
<tr>
<td><strong>[ FREQ &amp; DAMPING]</strong></td>
<td></td>
</tr>
<tr>
<td>[START FREQUENCY]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STARt 0 \sim 102400(HZ)}</td>
</tr>
<tr>
<td>[STOP FREQUENCY]</td>
<td>\texttt{CALCulate#:MARKer:BAND:STOP 0 \sim 102400(HZ)}</td>
</tr>
<tr>
<td>[COMPUTE OFF]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion OFF}</td>
</tr>
<tr>
<td>[COMPUTE COEFFICIENT]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion FREQuency}</td>
</tr>
<tr>
<td>[SUPPLEMENTAL INFO]</td>
<td>\texttt{CALCulate#:MARKer:FUNCtion SINF}</td>
</tr>
</tbody>
</table>
## System Group

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preset</strong></td>
<td></td>
</tr>
<tr>
<td><strong>BASIC</strong></td>
<td></td>
</tr>
<tr>
<td>[ DO PRESET ]</td>
<td>SYSTem:PRESet</td>
</tr>
<tr>
<td>[ RECALL AUTOSTATE ]</td>
<td>MMEMory:LOAD:STATe #,’&lt;FILENAME&gt;’</td>
</tr>
<tr>
<td>[ RUN PROGRAM 1 ]</td>
<td>PROGرام:NAME PROGرام1;STATe RUN</td>
</tr>
<tr>
<td>[ RUN PROGRAM 2 ]</td>
<td>PROGرام:NAME PROGرام2;STATe RUN</td>
</tr>
<tr>
<td>[ RUN PROGRAM 3 ]</td>
<td>PROGرام:NAME PROGرام3;STATe RUN</td>
</tr>
<tr>
<td>[ RUN PROGRAM 4 ]</td>
<td>PROGرام:NAME PROGرام4;STATe RUN</td>
</tr>
<tr>
<td>[ RUN PROGRAM 5 ]</td>
<td>PROGرام:NAME PROGرام5;STATe RUN</td>
</tr>
<tr>
<td>[ DISPLAY SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[ OFF ]</td>
<td>DISPlay:PROGرام[:MODE] OFF</td>
</tr>
<tr>
<td>[ FULL ]</td>
<td>DISPlay:PROGرام[:MODE] FULL</td>
</tr>
<tr>
<td>[ UPPER ]</td>
<td>DISPlay:PROGرام[:MODE] UPPer</td>
</tr>
<tr>
<td>[ LOWER ]</td>
<td>DISPlay:PROGرام[:MODE] LOWer</td>
</tr>
<tr>
<td>[ CLEAR SCREEN ]</td>
<td>DISPlay:ENABle OFF</td>
</tr>
<tr>
<td>[ CONTINUE ]</td>
<td>PROGرام:SELECTed:STATe CONTinue</td>
</tr>
<tr>
<td>[ INSTRUMNT BASIC ]</td>
<td></td>
</tr>
<tr>
<td>[RUN PROGRAM ]</td>
<td>PROGرام:SELECTed:STATе RUN</td>
</tr>
<tr>
<td>[SELECT PROGRAM ]</td>
<td></td>
</tr>
<tr>
<td>[ PROGRAM 1 ]</td>
<td>PROGرام:SELECTed;NAME PROG1</td>
</tr>
<tr>
<td>[ PROGRAM 2 ]</td>
<td>PROGرام:SELECTed;NAME PROG2</td>
</tr>
<tr>
<td>[ PROGRAM 3 ]</td>
<td>PROGرام:SELECTed;NAME PROG3</td>
</tr>
<tr>
<td>[ PROGRAM 4 ]</td>
<td>PROGرام:SELECTed;NAME PROG4</td>
</tr>
<tr>
<td>[ PROGRAM 5 ]</td>
<td>PROGرام:SELECTed;NAME PROG5</td>
</tr>
<tr>
<td>[LABEL PROGRAM ]</td>
<td>PROGرام:SELECTed;LABel ‘&lt;STRING&gt;’</td>
</tr>
<tr>
<td>[PRINT PROGRAM ]</td>
<td></td>
</tr>
<tr>
<td>[UTILITIES ]</td>
<td></td>
</tr>
<tr>
<td>[ MEMORY SIZE ]</td>
<td>PROGرام:SELECTed;MALLocate 1200 ~ 7340030</td>
</tr>
<tr>
<td>[ AUTO MEMORY ]</td>
<td>PROGرام:SELECTed;MALLocate DEFault</td>
</tr>
<tr>
<td>[ SCRATCH A ]</td>
<td></td>
</tr>
<tr>
<td>[ PERFORM SCRATCH ]</td>
<td>PROGرام:SELECTed;DELete[:SELECTed]</td>
</tr>
<tr>
<td>[DEBUG ]</td>
<td></td>
</tr>
<tr>
<td>[ RUN ]</td>
<td>PROGرام:SELECTed:STATе RUN</td>
</tr>
<tr>
<td>[ CONTINUE ]</td>
<td>PROGرام:SELECTed:STATе CONTinue</td>
</tr>
<tr>
<td>[ SINGLE STEP ]</td>
<td></td>
</tr>
<tr>
<td>[ LAST ERROR ]</td>
<td></td>
</tr>
<tr>
<td>[ EXAMINE VARIABLE ]</td>
<td>PROGرام:SELECTed;NUMBER</td>
</tr>
<tr>
<td></td>
<td>PROGرام:SELECTed;STRING</td>
</tr>
<tr>
<td>[ RESET ]</td>
<td>PROGرام:SELECTed;STATе STOP</td>
</tr>
</tbody>
</table>
### Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save/Recall</td>
<td></td>
</tr>
<tr>
<td>[ SAVE DATA ]</td>
<td></td>
</tr>
<tr>
<td>[SAVE TRACE]</td>
<td></td>
</tr>
<tr>
<td>[ INTO D1 ]</td>
<td>TRACe:DATA D1,TRACe#</td>
</tr>
<tr>
<td>[ INTO D2 ]</td>
<td>TRACe:DATA D2,TRACe#</td>
</tr>
<tr>
<td>[ INTO D3 ]</td>
<td>TRACe:DATA D3,TRACe#</td>
</tr>
<tr>
<td>[ INTO D4 ]</td>
<td>TRACe:DATA D4,TRACe#</td>
</tr>
<tr>
<td>[ INTO D5 ]</td>
<td>TRACe:DATA D5,TRACe#</td>
</tr>
<tr>
<td>[ INTO D6 ]</td>
<td>TRACe:DATA D6,TRACe#</td>
</tr>
<tr>
<td>[ INTO D7 ]</td>
<td>TRACe:DATA D7,TRACe#</td>
</tr>
<tr>
<td>[ INTO D8 ]</td>
<td>TRACe:DATA D8,TRACe#</td>
</tr>
<tr>
<td>[ INTO FILE ]</td>
<td>MMEMory:STORe:TRACe#, ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[SAVE CAPTURE]</td>
<td>MMEMory:STORe:TCAPture ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[SAVE WATERFALL]</td>
<td></td>
</tr>
<tr>
<td>[ INTO W1 ]</td>
<td>TRACe:WATerfall:DATA W1 TRACe#</td>
</tr>
<tr>
<td>[ INTO W2 ]</td>
<td>TRACe:WATerfall:DATA W2 TRACe#</td>
</tr>
<tr>
<td>[ INTO W3 ]</td>
<td>TRACe:WATerfall:DATA W3 TRACe#</td>
</tr>
<tr>
<td>[ INTO W4 ]</td>
<td>TRACe:WATerfall:DATA W4 TRACe#</td>
</tr>
<tr>
<td>[ INTO W5 ]</td>
<td>TRACe:WATerfall:DATA W5 TRACe#</td>
</tr>
<tr>
<td>[ INTO W6 ]</td>
<td>TRACe:WATerfall:DATA W6 TRACe#</td>
</tr>
<tr>
<td>[ INTO W7 ]</td>
<td>TRACe:WATerfall:DATA W7 TRACe#</td>
</tr>
<tr>
<td>[ INTO W8 ]</td>
<td>TRACe:WATerfall:DATA W8 TRACe#</td>
</tr>
<tr>
<td>[ INTO FILE ]</td>
<td>MMEMory:STORe::WATerfall TRACe#,, ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[CONTINUE SAVE ]</td>
<td>MMEMory:STORe:CONTinue</td>
</tr>
<tr>
<td>[ CATALOG ON OFF]</td>
<td>DISPlay:VIEW MMEMory</td>
</tr>
<tr>
<td>[SAVE STATE]</td>
<td>MMEMory:STORe:STATe #,’〈FILENAME〉’</td>
</tr>
<tr>
<td>[SAVE MORE]</td>
<td></td>
</tr>
<tr>
<td>[ SAVE UPPER LIM ]</td>
<td>MMEMory:STORe:LiMiT:UPPer:TRACe# ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE LOWER LIM ]</td>
<td>MMEMory:STORe:LiMiT:LOWer:TRACe# ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE MATH ]</td>
<td>MMEMory:STORe:MATh ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE PROGRAM ]</td>
<td>MMEMory:STORe:PROGram ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE PROGRAM ASCII BIN ]</td>
<td>MMEMory:STORe:PROGram:FORMat ASCii</td>
</tr>
<tr>
<td>[ SAVE FIT TABLE ]</td>
<td>MMEMory:STORe:CFIT ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE SNTH TABLE ]</td>
<td>MMEMory:STORe:SYNThesis ‘〈FILENAME〉’</td>
</tr>
<tr>
<td>[ SAVE AUTOSTATE ]</td>
<td>MMEMory:STORe:STATe 1, ‘NVRAM:AUTO _ST’</td>
</tr>
</tbody>
</table>
# Front Panel Key to GPIB Command Cross-Reference

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ RECALL DATA ]</td>
<td>[ RECALL TRACE ]</td>
</tr>
<tr>
<td>[ FROM FILE INTO D1 ]</td>
<td>MMEMory:LOAD:TRACe D1, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO D2 ]</td>
<td>MMEMory:LOAD:TRACe D2, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO D4 ]</td>
<td>MMEMory:LOAD:TRACe D4, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO D5 ]</td>
<td>MMEMory:LOAD:TRACe D5, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO D6 ]</td>
<td>MMEMory:LOAD:TRACe D6, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO D8 ]</td>
<td>MMEMory:LOAD:TRACe D8, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL CAPTURE ]</td>
<td>MMEMory:LOAD:TCAPture '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL WATERFALL ]</td>
<td>[ FROM FILE INTO W1 ]</td>
</tr>
<tr>
<td></td>
<td>MMEMory:LOAD:WATerfall W1, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO W2 ]</td>
<td>MMEMory:LOAD:WATerfall W2, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO W5 ]</td>
<td>MMEMory:LOAD:WATerfall W5, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO W6 ]</td>
<td>MMEMory:LOAD:WATerfall W6, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO W7 ]</td>
<td>MMEMory:LOAD:WATerfall W7, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ FROM FILE INTO W8 ]</td>
<td>MMEMory:LOAD:WATerfall W8, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ CONTINUE RECALL ]</td>
<td>MMEMory:LOAD:CONTinue</td>
</tr>
<tr>
<td>[ RECALL STATE ]</td>
<td>MMEMory:LOAD:STATe #, '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL MORE ]</td>
<td>[ RECALL UPPER LIM ]</td>
</tr>
<tr>
<td></td>
<td>MMEMory:LOAD:LIMit:UPPer:TRACe# '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL LOWER LIM ]</td>
<td>MMEMory:LOAD:LIMit:LOWer:TRACe# '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL MATH ]</td>
<td>MMEMory:LOAD:MATH '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL PROGRAM ]</td>
<td>MMEMory:LOAD:PROGram '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RCL FIT TABLE ]</td>
<td>MMEMory:LOAD:CFIT '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RCL SYNTH TABLE ]</td>
<td>MMEMory:LOAD:SYNThesis '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ RECALL AUTOSTATE ]</td>
<td>MMEMory:LOAD:STATe 1, 'NVRAM:AUTO_ST'</td>
</tr>
<tr>
<td>[ DEFAULT DISK ]</td>
<td>[ NON-VOL RAM DISK ]</td>
</tr>
<tr>
<td></td>
<td>MMEMory:MSIS 'NVRAM:'</td>
</tr>
<tr>
<td>[ VOLATILE RAM DISK ]</td>
<td>MMEMory:MSIS 'RAM:'</td>
</tr>
<tr>
<td>[ INTERNAL DISK ]</td>
<td>MMEMory:MSIS 'INT:'</td>
</tr>
<tr>
<td>[ EXTERNAL DISK ]</td>
<td>MMEMory:MSIS 'EXT:'</td>
</tr>
<tr>
<td>[ CREATE DIRECTORY ]</td>
<td>MMEMory:MDIRECTORY '&lt;STRING&gt;'</td>
</tr>
<tr>
<td>[ DELETE DIRECTORY ]</td>
<td>MMEMory:DELe te '&lt;MMEMNAME&gt;'</td>
</tr>
</tbody>
</table>
Cross-Reference from Front-Panel Keys to GPIB Commands

### Disk Utilities

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ RENAME FILE ]</td>
<td>MME:REName '&lt;FILENAME&gt;', '&lt;FILE&gt;'</td>
</tr>
<tr>
<td>[ ORIGINAL FILENAME ]</td>
<td>&lt;FILENAME&gt;</td>
</tr>
<tr>
<td>[ NEW FILENAME ]</td>
<td>&lt;FILE&gt;</td>
</tr>
<tr>
<td>[ PERFORM RENAME ]</td>
<td>MMEM:REName '&lt;FILENAME&gt;', '&lt;FILE&gt;'</td>
</tr>
<tr>
<td>[ DELETE FILE ]</td>
<td>MMEM:DELe ' &lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ DELETE ALL FILES ]</td>
<td>MMEM:DELe '&lt;DISK&gt;'</td>
</tr>
<tr>
<td>[ COPY FILE ]</td>
<td>MMEM:COPY '&lt;FILENAME&gt;', '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ SOURCE FILENAME ]</td>
<td>&lt;FILENAME&gt;</td>
</tr>
<tr>
<td>[ DESTIN FILENAME ]</td>
<td>&lt;FILENAME&gt;</td>
</tr>
<tr>
<td>[ PERFORM FILE COPY ]</td>
<td>MMEM:COPY '&lt;FILENAME&gt;', '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ COPY ALL FILES ]</td>
<td>MMEM:COPY '&lt;DISK&gt;', '&lt;DISK&gt;'</td>
</tr>
<tr>
<td>[ SOURCE DISK ]</td>
<td>&lt;DISK&gt;</td>
</tr>
<tr>
<td>[ DESTIN DISK ]</td>
<td>&lt;DISK&gt;</td>
</tr>
<tr>
<td>[ PERFORM COPY ALL ]</td>
<td>MMEM:COPY '&lt;DISK&gt;', '&lt;DISK&gt;'</td>
</tr>
<tr>
<td>[ FORMAT DISK ]</td>
<td>MMEM:INITialize '&lt;DISK&gt;', [ LIF</td>
</tr>
<tr>
<td>[ DEFAULT DISK ]</td>
<td>MMEM:MSIS '&lt;DISK&gt;'</td>
</tr>
</tbody>
</table>

### Local/GPIB

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ABORT GPIB ]</td>
<td>ABORt</td>
</tr>
<tr>
<td>[ ANALYZER ADDRESS ]</td>
<td>SYSTEM:COMM:GPIB[:SELF]:ADDRes 0 ~ 30</td>
</tr>
<tr>
<td>[ GPIB ECHO ON OFF ]</td>
<td>DIS:GPIB:ECHO ON</td>
</tr>
<tr>
<td>[ PLOTTER ADDRESS ]</td>
<td>HCOPY:ADDRes 0 ~ 30</td>
</tr>
<tr>
<td>[ PRINTER ADDRESS ]</td>
<td>HCOPY:PRINT:ADDRes 0 ~ 30</td>
</tr>
<tr>
<td>[ DISK ADDRESS ]</td>
<td>MMEM:DISK:ADDRes 0 ~ 30</td>
</tr>
</tbody>
</table>
### Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ START PLOT/PRNT ]</td>
<td>HCOPy[:IMMediate]</td>
</tr>
<tr>
<td>[ PLOT/PRINT DEVICE ]</td>
<td></td>
</tr>
<tr>
<td>[HP-GL PLOTTER]</td>
<td>HCOPy:DEViCe:LANGuage HPGL</td>
</tr>
<tr>
<td>[RASTER PRINTER]</td>
<td>HCOPy:DEViCe:LANGuage PCL</td>
</tr>
<tr>
<td>[HP-GL PRINTER]</td>
<td>HCOPy:DEViCe:LANGuage PHPG</td>
</tr>
<tr>
<td>[ PLOT/PRINT DESTINATION ]</td>
<td></td>
</tr>
<tr>
<td>[OUTPUT TO GPIB]</td>
<td>HCOPy:DESTination 'SYSTem:COMMunicate:GPIB:RDEV'</td>
</tr>
<tr>
<td>[OUTPUT TO SERIAL]</td>
<td>HCOPy:DESTination 'SYSTem:COMMunicate:SErial'</td>
</tr>
<tr>
<td>[OUTPUT TO PARALLEL]</td>
<td>HCOPy:DESTination 'SYSTem:COMMunicate:CENTronics'</td>
</tr>
<tr>
<td>[OUTPUT TO FILE]</td>
<td>HCOPy:DESTination 'MMEM'</td>
</tr>
<tr>
<td>[OUTPUT FILENAME]</td>
<td>MMEMory:NAME '&lt;FILENAME&gt;'</td>
</tr>
<tr>
<td>[ PLOT DATA SELECT ]</td>
<td></td>
</tr>
<tr>
<td>[ALL]</td>
<td>HCOPy:ITEM:ALL[:IMMediate]</td>
</tr>
<tr>
<td>[TRACE]</td>
<td>HCOPy:ITEM:[WINDow#]:TRACe[:IMMediate]</td>
</tr>
<tr>
<td>[TRACE MARKER]</td>
<td>HCOPy:ITEM:[WINDow#]:TRACe:MARKer[:IMMediate]</td>
</tr>
<tr>
<td>[MARKER REFERENCE]</td>
<td>HCOPy:ITEM:[WINDow#]:TRACe:MARKer:REference[:IMMediate]</td>
</tr>
<tr>
<td>[GRID]</td>
<td>HCOPy:ITEM:[WINDow#]:TRACe:GRAticule[:IMMediate]</td>
</tr>
<tr>
<td>[ PLOT PEN SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[DEFAULT PENS]</td>
<td>HCOPy:COlor:DEFault</td>
</tr>
<tr>
<td>[TRACE A PEN]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE B PEN]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE C PEN]</td>
<td>HCOPy:ITEM:WINDow3:TRACe:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE D PEN]</td>
<td>HCOPy:ITEM:WINDow4:TRACe:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[ MARKER PEN SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[TRACE A MKR PEN]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:MARKer:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE B MKR PEN]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:MARKer:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE C MKR PEN]</td>
<td>HCOPy:ITEM:WINDow3:TRACe:MARKer:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[TRACE D MKR PEN]</td>
<td>HCOPy:ITEM:WINDow4:TRACe:MARKer:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[ALPHA PEN]</td>
<td>HCOPy:ITEM:LABel:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[GRID PEN]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:GRAticule:COlor 0 ~ 16</td>
</tr>
<tr>
<td>[ PLOT LINE SETUP ]</td>
<td></td>
</tr>
<tr>
<td>[TRACE A LINE TYPE]</td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:LTYPe SOLId</td>
</tr>
<tr>
<td>[DOTTED]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:LTYPe DOTTed</td>
</tr>
<tr>
<td>[DASHED]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:LTYPe DASHed</td>
</tr>
<tr>
<td>[USER LINE TYPE]</td>
<td>HCOPy:ITEM:WINDow1:TRACe:LTYPe -6 ~ 6</td>
</tr>
<tr>
<td>[TRACE B LINE TYPE]</td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:LTYPe SOLId</td>
</tr>
<tr>
<td>[DOTTED]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:LTYPe DOTTed</td>
</tr>
<tr>
<td>[DASHED]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:LTYPe DASHed</td>
</tr>
<tr>
<td>[USER LINE TYPE]</td>
<td>HCOPy:ITEM:WINDow2:TRACe:LTYPe -6 ~ 6</td>
</tr>
</tbody>
</table>
### Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRACE C LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDOW3:TRACE:LTYPe SOLid</td>
</tr>
<tr>
<td>[DOTTED]</td>
<td>HCOPy:ITEM:WINDOW3:TRACE:LTYPe DOTted</td>
</tr>
<tr>
<td><strong>TRACE D LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDOW4:TRACE:LTYPe SOLid</td>
</tr>
<tr>
<td>[DOTTED]</td>
<td>HCOPy:ITEM:WINDOW4:TRACE:LTYPe DOTted</td>
</tr>
<tr>
<td><strong>LIMIT A LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDOW1:TRACE:LIMIT:LTYPe SOLid</td>
</tr>
<tr>
<td>[DOTTED]</td>
<td>HCOPy:ITEM:WINDOW1:TRACE:LIMIT:LTYPe DOTted</td>
</tr>
<tr>
<td><strong>LIMIT B LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LIMIT C LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LIMIT D LINE TYPE</strong></td>
<td></td>
</tr>
<tr>
<td>[SOLID]</td>
<td>HCOPy:ITEM:WINDOW4:TRACE:LIMIT:LTYPe SOLid</td>
</tr>
<tr>
<td><strong>MORE SETUP</strong></td>
<td></td>
</tr>
<tr>
<td>[PLOT PEN SPEED]</td>
<td></td>
</tr>
<tr>
<td>[FAST 50 cm/s]</td>
<td>HCOPy:DEVice:SPEed 50</td>
</tr>
<tr>
<td>[SLOW 10 cm/s]</td>
<td>HCOPy:DEVice:SPEed 10</td>
</tr>
<tr>
<td>[DEFINE ? cm/s]</td>
<td>HCOPy:DEVice:SPEed 1 ~ 100</td>
</tr>
<tr>
<td><strong>P1 P2 SETUP</strong></td>
<td></td>
</tr>
<tr>
<td>[USER P1 P2 ON OFF]</td>
<td>HCOPy:PAGE:DIM:Auto OFF</td>
</tr>
<tr>
<td>[USER P1 X]</td>
<td></td>
</tr>
<tr>
<td>[USER P2 X]</td>
<td></td>
</tr>
<tr>
<td><strong>TIME STMP ON OFF</strong></td>
<td></td>
</tr>
<tr>
<td>[PAGE EJCT ON OFF]</td>
<td>HCOPy:EJECt OFF</td>
</tr>
<tr>
<td><strong>TITLE LINE 1</strong></td>
<td>HCOPy:TITL1 ‘&lt;STRING&gt;’</td>
</tr>
<tr>
<td><strong>TITLE LINE 2</strong></td>
<td>HCOPy:TITL2 ‘&lt;STRING&gt;’</td>
</tr>
</tbody>
</table>
## Front Panel Key to GPIB Commands

<table>
<thead>
<tr>
<th>Serial Setup</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ BAUD RATE ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:BAUD 300 ~ 9600</td>
</tr>
<tr>
<td>[ BITS/CHAR ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:BITS 5 ~ 8</td>
</tr>
<tr>
<td>[ STOP BITS ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:SBitS 1 ~ 2</td>
</tr>
<tr>
<td>[ PARITY ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:PARITY[:TYPE] NONE</td>
</tr>
<tr>
<td>[ PRTY CHK ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:PARITY:CHECk OFF</td>
</tr>
<tr>
<td>[ RCVR PACE ]</td>
<td>SYSTem:COMMunicate:SERial[:RECeive]:PACE NONE</td>
</tr>
</tbody>
</table>

### System

**Utility**

<table>
<thead>
<tr>
<th>Show Last Messages</th>
<th>Display:VIEW MESS</th>
</tr>
</thead>
</table>

**Calibratn**

<table>
<thead>
<tr>
<th>Auto Cal On Off</th>
<th>CALibration:AUTO OFF</th>
<th>0</th>
<th>ON</th>
<th>1</th>
<th>ONCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Cal</td>
<td>CALibration:AUTO ONCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Save Ch1 Cal Trace

<table>
<thead>
<tr>
<th>Into D1</th>
<th>TRACe:DATA D1,CAL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Into D2</td>
<td>TRACe:DATA D2,CAL1</td>
</tr>
<tr>
<td>Into D3</td>
<td>TRACe:DATA D3,CAL1</td>
</tr>
<tr>
<td>Into D4</td>
<td>TRACe:DATA D4,CAL1</td>
</tr>
<tr>
<td>Into D5</td>
<td>TRACe:DATA D5,CAL1</td>
</tr>
<tr>
<td>Into D6</td>
<td>TRACe:DATA D6,CAL1</td>
</tr>
<tr>
<td>Into D7</td>
<td>TRACe:DATA D7,CAL1</td>
</tr>
<tr>
<td>Into D8</td>
<td>TRACe:DATA D8,CAL1</td>
</tr>
</tbody>
</table>

#### Save Ch2 Cal Trace

<table>
<thead>
<tr>
<th>Into D1</th>
<th>TRACe:DATA D1,CAL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Into D2</td>
<td>TRACe:DATA D2,CAL2</td>
</tr>
<tr>
<td>Into D3</td>
<td>TRACe:DATA D3,CAL2</td>
</tr>
<tr>
<td>Into D4</td>
<td>TRACe:DATA D4,CAL2</td>
</tr>
<tr>
<td>Into D5</td>
<td>TRACe:DATA D5,CAL2</td>
</tr>
<tr>
<td>Into D6</td>
<td>TRACe:DATA D6,CAL2</td>
</tr>
<tr>
<td>Into D7</td>
<td>TRACe:DATA D7,CAL2</td>
</tr>
<tr>
<td>Into D8</td>
<td>TRACe:DATA D8,CAL2</td>
</tr>
</tbody>
</table>

#### Save Ch3 Cal Trace

<table>
<thead>
<tr>
<th>Into D1</th>
<th>TRACe:DATA D1,CAL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Into D2</td>
<td>TRACe:DATA D2,CAL3</td>
</tr>
<tr>
<td>Into D3</td>
<td>TRACe:DATA D3,CAL3</td>
</tr>
<tr>
<td>Into D4</td>
<td>TRACe:DATA D4,CAL3</td>
</tr>
<tr>
<td>Into D5</td>
<td>TRACe:DATA D5,CAL3</td>
</tr>
<tr>
<td>Into D6</td>
<td>TRACe:DATA D6,CAL3</td>
</tr>
<tr>
<td>Into D7</td>
<td>TRACe:DATA D7,CAL3</td>
</tr>
<tr>
<td>Into D8</td>
<td>TRACe:DATA D8,CAL3</td>
</tr>
</tbody>
</table>
Cross-Reference from Front-Panel Keys to GPIB Commands

<table>
<thead>
<tr>
<th>Front Panel Key</th>
<th>GPIB Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SAVE CH4 CAL TRACE ]</td>
<td>TRACE:DATA D1,CAL4</td>
</tr>
<tr>
<td>[ INTO D1 ]</td>
<td>TRACE:DATA D2,CAL4</td>
</tr>
<tr>
<td>[ INTO D2 ]</td>
<td>TRACE:DATA D3,CAL4</td>
</tr>
<tr>
<td>[ INTO D3 ]</td>
<td>TRACE:DATA D4,CAL4</td>
</tr>
<tr>
<td>[ INTO D4 ]</td>
<td>TRACE:DATA D5,CAL4</td>
</tr>
<tr>
<td>[ INTO D5 ]</td>
<td>TRACE:DATA D6,CAL4</td>
</tr>
<tr>
<td>[ INTO D6 ]</td>
<td>TRACE:DATA D7,CAL4</td>
</tr>
<tr>
<td>[ INTO D7 ]</td>
<td>TRACE:DATA D8,CAL4</td>
</tr>
<tr>
<td>[ INTO D8 ]</td>
<td>SYSTEM:BEEP:STATE OFF</td>
</tr>
<tr>
<td>[ BEEPER ON OFF ]</td>
<td>SYSTEM:TIME 0 ~ 2359</td>
</tr>
<tr>
<td>[ TIME HHMM ]</td>
<td>SYSTEM:DATE 10100 ~ 123199</td>
</tr>
<tr>
<td>[ DATE MMDDYY ]</td>
<td>HCOPY:TSTamp:MODE CHOice1</td>
</tr>
<tr>
<td>[ TIMESTAMP SETUP ]</td>
<td>HCOPY:TSTamp:MODE CHOice2</td>
</tr>
<tr>
<td>[ 24 HR DD/MM/YY ]</td>
<td>HCOPY:TSTamp:MODE CHOice3</td>
</tr>
<tr>
<td>[ 24 HR DD.MM.YY ]</td>
<td>HCOPY:TSTamp:MODE CHOice4</td>
</tr>
<tr>
<td>[ 24 HR YY MM DD ]</td>
<td>HCOPY:TSTamp:MODE CHOice5</td>
</tr>
<tr>
<td>[ 12 HR DD/MM/YY ]</td>
<td>DISPLAY:VIEW OPT</td>
</tr>
<tr>
<td>[ OPTIONS SETUP ]</td>
<td>DISPLAY:VIEW MEM</td>
</tr>
<tr>
<td>[ REMOVE CAPTURE ]</td>
<td>MEMory:DElete[:NAME] WATERfall</td>
</tr>
<tr>
<td>[ REMOVE WATERFALL ]</td>
<td>MEMory:DElete[:NAME] WREGister</td>
</tr>
<tr>
<td>[ REMOVE WTFL REGS ]</td>
<td>MEMory:DElete[:NAME] PROgram</td>
</tr>
<tr>
<td>[ REMOVE PROGRAMS ]</td>
<td>MEMory:DElete[:NAME] RDISK</td>
</tr>
<tr>
<td>[ REMOVE RAM DISK ]</td>
<td>SYSTEM:FAN OFF</td>
</tr>
<tr>
<td>[ REMOVE RAM DISK ]</td>
<td>SYSTEM:FAN FULL</td>
</tr>
<tr>
<td>[ FAN SETUP ]</td>
<td>SYSTEM:FAN AUTO</td>
</tr>
<tr>
<td>[ FAN OFF ]</td>
<td>SYSTEM:FAN AUTO</td>
</tr>
<tr>
<td>[ FAN FULL SPEED ]</td>
<td>SYSTEM:FAN AUTO</td>
</tr>
<tr>
<td>[ AUTOMATIC SPEED ]</td>
<td>SYSTEM:FAN AUTO</td>
</tr>
<tr>
<td>[ MORE ]</td>
<td>SYSTEM:FLOG:CLEar</td>
</tr>
<tr>
<td>[ FAULT LOG ]</td>
<td>DISPLAY:VIEW FTABle</td>
</tr>
<tr>
<td>[ CLEAR FAULT LOG ]</td>
<td>SYSTem:FLOG:CLEar</td>
</tr>
<tr>
<td>[ SELF TEST ]</td>
<td>*TST?</td>
</tr>
<tr>
<td>[ QUICK CONF TEST ]</td>
<td>TEST:LONG</td>
</tr>
<tr>
<td>[ LONG CONF TEST ]</td>
<td>DISPLAY:VIEW TTAB</td>
</tr>
<tr>
<td>[ TEST LOG ]</td>
<td>TEST:LOG:CLEar</td>
</tr>
<tr>
<td>[ CLEAR TEST LOG ]</td>
<td>*IDN?</td>
</tr>
</tbody>
</table>
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.
    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.

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.
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.
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.
dbSPL 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.
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.**

-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

<table>
<thead>
<tr>
<th>GPIB Command</th>
<th>Instrument Mode (INST:SEL)</th>
<th>FFT</th>
<th>OCT</th>
<th>ORD</th>
<th>SINE</th>
<th>CORR</th>
<th>HIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*CLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*ESE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*ESR?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*IDN?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*OPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*OPC?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*OPT?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*PCB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*PSC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*RST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*SRE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*STB?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*TRG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*TST?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*WAI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:IMMediate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:RPM:INCRement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:RPM:MODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:RPM:THReshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:SOURrce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM:TIMer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCulate:ACTive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:ABORt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:CPy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Instrument Modes

<table>
<thead>
<tr>
<th>GPIB Command</th>
<th>Instrument Mode (INST:SEL )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>CALCulate:CFIT:DESTination</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:FREQuency[:AUTO]</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:FREQuency:STARt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:FREQuency:STOP</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:FSCale</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:ORDer:AUTO</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:ORDer:POLes</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:ORDer:ZERos</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:TDELay</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:WEIGht:AUTO</td>
<td></td>
</tr>
<tr>
<td>CALCulate:CFIT:WEIGht:REGister</td>
<td></td>
</tr>
<tr>
<td>CALCulate:DATA?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:DATA:HEADer:POINts?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:FEED</td>
<td></td>
</tr>
<tr>
<td>CALCulate:FORMat</td>
<td></td>
</tr>
<tr>
<td>CALCulate:GDAPerture:APERture</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:LIMic:BEEp[:;STATe]</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:FAIL?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:CLEar:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:MOVE:Y</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:REPort:DATA?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:REPort:YDATA?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:SEGMENT</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:SEGMENT:CLEar</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:LOWer:TRACe:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:STATe</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:CLEar:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:MOVE:Y</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:REPort:DATA?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:REPort:YDATA?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:SEGMENT</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:SEGMENT:CLEar</td>
<td></td>
</tr>
<tr>
<td>CALCulate:LIMic:UPPer:TRACe:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:BAND:STARt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:BAND:STOP</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:COUPled:STATe</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABLE:CLEar:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABLE:COPY</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABLE:DATA</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABLE:X:DATA</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL )</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABle:X:DELete</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABle:X:INSert</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABle:X:LABel</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:DTABle:X:SELeCt:POINt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:FUNCtion</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:FUNCtion:RESul?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:HARMonic:COUNt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:HARMonic:FUNDamental</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:MAXimum:GLOBal</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:MAXimum:GLOB:TRACk</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:MAXimum:LEFT</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:MAXimum:RIGHT</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:MODE</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:POSItion</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:POSItion:POINt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:REFERence:X</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:REFERence:Y</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:SIDeband:CARRier</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:SIDeband:COUNt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:SIDeband:INCRement</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:STATe</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:X:ABSolute</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:X:RELative</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:Y:ABSolute?</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MARKer:Y:RELative</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MATH:CONSTant</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MATH:DATA</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MATH:EXPRession</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MATH:SELeCt</td>
<td></td>
</tr>
<tr>
<td>CALCulate:MATH:STATe</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:COPY</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:DATA</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:DESTination</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:FScale</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:GAIN</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:IMMediate</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:SPACing</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:TDELay</td>
<td></td>
</tr>
<tr>
<td>CALCulate:SYNthesis:TTYPe</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:AMPLitude</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:ANGLe</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>CALCulate:UNIT:DBReference</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:DBReference:IMPedance</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:DBReference:USER:LABel</td>
<td></td>
</tr>
<tr>
<td>CALCULATE:UNIT:DBReference:USER:REference</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:MECHanical</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:VOLTage</td>
<td></td>
</tr>
<tr>
<td>CALCulate:UNIT:X</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:UNIT:X:ORDER:FACTor</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:UNIT:X:USER:FREQuency:FACTor</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:UNIT:X:USER:FREQuency:LABel</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:UNIT:X:USER:TIME:FACTor</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:UNIT:X:USER:TIME:LABel</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:COUNt</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:DATA</td>
<td></td>
</tr>
<tr>
<td>CALCulate:WATerfall:SLICe:COPY</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:SLICe:SELect</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:SLICe:SELect:POINt</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:TRACe:COPY</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:TRACe:SELect</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:WATerfall:TRACe:SELect:POINt</td>
<td>not valid</td>
</tr>
<tr>
<td>CALCulate:X:DATA</td>
<td></td>
</tr>
<tr>
<td>CALLibration:ALL?</td>
<td></td>
</tr>
<tr>
<td>CALLibration:AUTO</td>
<td></td>
</tr>
<tr>
<td>DISPlay:ANNnotation:ALL</td>
<td></td>
</tr>
<tr>
<td>DISPlay:BODE</td>
<td>not valid</td>
</tr>
<tr>
<td>DISPlay:BRIGHTness</td>
<td></td>
</tr>
<tr>
<td>DISPlay:ERROR</td>
<td></td>
</tr>
<tr>
<td>DISPlay:EXTERNAL:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:FORMAT</td>
<td></td>
</tr>
<tr>
<td>DISPlay:GPIB:ECCHO</td>
<td></td>
</tr>
<tr>
<td>DISPlay:PROGram:KEY:BOX</td>
<td></td>
</tr>
<tr>
<td>DISPlay:PROGram:KEY:BRACket</td>
<td></td>
</tr>
<tr>
<td>DISPlay:PROGram:MODE</td>
<td></td>
</tr>
<tr>
<td>DISPlay:PROGram:VECTor:BUFFER:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:RPM:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:STATe</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL )</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>DISPlay:TCAPture:ENVelope:STATe</td>
<td>FFT OCT ORD SINE CORR HIST</td>
</tr>
<tr>
<td>DISPlay:VIEW</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:DATAle:MARKer:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:DATAle:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:LIMit:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:POLar:CLOCkwise</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:POLar:ROTation</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:APOWer:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:BPoWer:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:GRAT:GRID:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:LABel</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRAC:LABel:DEF:STATe</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:X:MATch</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:X:SCALe:AUTO</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:X:SCALe:LEFT</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:X:SCALe:RIGHt</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:X:SPACing</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:Y:MATch</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:Y:SCALe:AUTO</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRAC,Y:SCAL:BOTTom</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:Y:SCAL:CENTER</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRAC,Y:SCAL:PDIVision</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRAC,Y:SCAL:REFERENCE</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:Y:SCALe:TOP</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:TRACe:Y:SPACing</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:BASEline</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:COUNt</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:HEIGHT</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:HIDDEN</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:SKEW</td>
<td></td>
</tr>
<tr>
<td>DISPlay:WINDow:WATerfall:SKEW:ANGLE</td>
<td></td>
</tr>
<tr>
<td>FORMat:DATA</td>
<td></td>
</tr>
<tr>
<td>HCOPy:COlor:DEFaultn</td>
<td></td>
</tr>
<tr>
<td>HCOPy:DESTination</td>
<td></td>
</tr>
<tr>
<td>HCOPy:DEVice:LANGuage</td>
<td></td>
</tr>
<tr>
<td>HCOPy:DEVice:SPEed</td>
<td></td>
</tr>
<tr>
<td>HCOPy:IMMediate</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:ALL:IMMediate</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:FFEed:STATe</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:LABel:COlor</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:LABel:STATe</td>
<td></td>
</tr>
</tbody>
</table>

**Invalid Commands:**
- DISPlay:WINDow:WATerfall:BASeline
- DISPlay:WINDow:WATerfall:COUNt
- DISPlay:WINDow:WATerfall:HEIGHT
- DISPlay:WINDow:WATerfall:HIDDEN
- DISPlay:WINDow:WATerfall:SKEW
- DISPlay:WINDow:WATerfall:SKEW:ANGLE

**Valid Commands:**
- 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
<table>
<thead>
<tr>
<th>GPIB Command</th>
<th>Instrument Mode (INST:SEL)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>HCOPy:ITEM:LABel:TEXT</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:TDSamp:FORMat</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:TDSamp:STATe</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:COLor</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:GRT:COLor</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:GRT:IMMediate</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:IMMediate</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:LTYPe</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:LTYPe</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WINDow:TRAc:MARK:COLor</td>
<td></td>
</tr>
<tr>
<td>HCOPy:ITEM:WIND:TRAc:MARK:REF:IMM</td>
<td></td>
</tr>
<tr>
<td>HCOPy:PAGE:DIMensions:USER:LEFT</td>
<td></td>
</tr>
<tr>
<td>HCOPy:PAGE:DIMensions:USER:RIGHT</td>
<td></td>
</tr>
<tr>
<td>HCOPy:TITLE</td>
<td></td>
</tr>
<tr>
<td>INITiate:CONTinuous</td>
<td></td>
</tr>
<tr>
<td>INITiate:IMMediate</td>
<td></td>
</tr>
<tr>
<td>INPut:BIAS:STATe</td>
<td></td>
</tr>
<tr>
<td>INPut:COUPling</td>
<td></td>
</tr>
<tr>
<td>INPut:FILTer:AEWighting:STATe</td>
<td></td>
</tr>
<tr>
<td>INPut:FILTer:LPASs:STATe</td>
<td></td>
</tr>
<tr>
<td>INPut:LOW</td>
<td></td>
</tr>
<tr>
<td>INPut:REFerence:DIRection</td>
<td></td>
</tr>
<tr>
<td>INPut:REFerence:POINt</td>
<td></td>
</tr>
<tr>
<td>INPut:STATe</td>
<td></td>
</tr>
<tr>
<td>INSTrument:NSElec</td>
<td></td>
</tr>
<tr>
<td>INSTrument:SELECT</td>
<td></td>
</tr>
<tr>
<td>MEMory:CATalog:ALL</td>
<td></td>
</tr>
<tr>
<td>MEMory:CATalog:NAME</td>
<td></td>
</tr>
<tr>
<td>MEMory:DELete:ALL</td>
<td></td>
</tr>
<tr>
<td>MEMory:DELete:NAME</td>
<td></td>
</tr>
<tr>
<td>MEMory:FREE:ALL</td>
<td></td>
</tr>
<tr>
<td>MMEMory:COPY</td>
<td></td>
</tr>
<tr>
<td>MMEMory:DELete</td>
<td></td>
</tr>
<tr>
<td>MMEMory:DISK:ADDRess</td>
<td></td>
</tr>
<tr>
<td>MMEMory:DISK:UNIT</td>
<td></td>
</tr>
<tr>
<td>MMEMory:FSYSTEM</td>
<td></td>
</tr>
<tr>
<td>MMEMory:INITialize</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL)</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>MMEMory:LOAD:CFIT</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:CONTinue</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:DTABLE:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:LIMIT:LOWer:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:LIMIT:UPPer:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:MATH</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:PROGRAM</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:STATE</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:SYNThesis</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:TCAPture</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:LOAD:WATERfall</td>
<td></td>
</tr>
<tr>
<td>MMEMory:MOVE</td>
<td></td>
</tr>
<tr>
<td>MMEMory:MSIS</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:CFIT</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:CONTinue</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:DTABLE:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:LIMIT:LOWer:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:LIMIT:UPPer:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:MATH</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:PROGRAM</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:STATE</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:SYNThesis</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:TCAPture</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:TRACe</td>
<td></td>
</tr>
<tr>
<td>MMEMory:STORE:WATERfall</td>
<td></td>
</tr>
<tr>
<td>OUTPut:STATE</td>
<td></td>
</tr>
<tr>
<td>PROGram:EDIT:ENABLE</td>
<td></td>
</tr>
<tr>
<td>PROGram:EXPLICIT:DEFINE</td>
<td></td>
</tr>
<tr>
<td>PROGram:EXPLICIT:LABEL</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:DEFINE</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:DELETE:ALL</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:DELETE:SELECTED</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:LABEL</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:NAME</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:NUMBER</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:STATE</td>
<td></td>
</tr>
<tr>
<td>PROGram:SELECTED:STRING</td>
<td></td>
</tr>
<tr>
<td>SENSE:AVERAGE:CONFidence</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVERAGE:COUNT</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL )</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>SENSE:AVerage:HOLD</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:IMpulse</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:IRES:RATE</td>
<td></td>
</tr>
<tr>
<td>SENSE:AVerage:IRES:STATE</td>
<td></td>
</tr>
<tr>
<td>SENSE:AVerage:PREView</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:PREView:ACCept</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:PREView:REJect</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:PREView:TIME</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:STATe</td>
<td></td>
</tr>
<tr>
<td>SENSE:AVerage:TCONtrol</td>
<td></td>
</tr>
<tr>
<td>SENSE:AVerage:TIME</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:AVerage:TYPE</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FEED</td>
<td></td>
</tr>
<tr>
<td>SENSE:FREQuency:BLOCksiz</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:CENTer</td>
<td></td>
</tr>
<tr>
<td>SENSE:FREQuency:MANual</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:RESolution</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:RES:Solution:AUTO</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:RES:Solution:MCHange</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:RES:Solution:MINimum</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:RESolution:OCTave</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:SPAN</td>
<td></td>
</tr>
<tr>
<td>SENSE:FREQuency:SPAN:FULL</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:SPAN:LINK</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:STARt</td>
<td></td>
</tr>
<tr>
<td>SENSE:FREQuency:STEP:INCRement</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:FREQuency:STOP</td>
<td></td>
</tr>
<tr>
<td>SENSE:HISTogram:BINS</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:MAXimum</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:RESolution</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:RESolution:TRACk</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:RPM:MAXimum</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:RPM:MINimum</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:ORDer:TRACk</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:REFerence</td>
<td></td>
</tr>
<tr>
<td>SENSE:REJect:STATe</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:SWEep:DIRection</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:SWEep:DWELI</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:SWEep:MODE</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:SWEep:OVERlap</td>
<td>not valid</td>
</tr>
<tr>
<td>SENSE:SWEep:SPACing</td>
<td>not valid</td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL)</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td></td>
<td>FFT</td>
</tr>
<tr>
<td>SENSe:SWep:STIMe</td>
<td></td>
</tr>
<tr>
<td>SENSe:SWep:TIME</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:ABORt</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:DELe</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:IMMediate</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:LENGTH</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:MA</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:STARt</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:STOP</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:TACHomer:RPM:MAX</td>
<td></td>
</tr>
<tr>
<td>SENSe:TCap:TACHomer:STATe</td>
<td></td>
</tr>
<tr>
<td>SENSe:VOLT:DC:RANG:AU</td>
<td></td>
</tr>
<tr>
<td>SENSe:VOLT:DC:RANG:AU:DIRection</td>
<td></td>
</tr>
<tr>
<td>SENSe:VOLT:DC:RANG:UNIT:MODE</td>
<td></td>
</tr>
<tr>
<td>SENSe:VOLT:DC:RANG:UPPer</td>
<td></td>
</tr>
<tr>
<td>SENSe:WIND:EXP</td>
<td></td>
</tr>
<tr>
<td>SENSe:WIND:FORCe</td>
<td></td>
</tr>
<tr>
<td>SENSe:WIND:ORDER:DC</td>
<td></td>
</tr>
<tr>
<td>SENSe:WIND:TYPE</td>
<td></td>
</tr>
<tr>
<td>SOUR:BUSt</td>
<td></td>
</tr>
<tr>
<td>SOUR:FRQ:CW</td>
<td></td>
</tr>
<tr>
<td>SOUR:FRQ:FIXed</td>
<td></td>
</tr>
<tr>
<td>SOUR:FUNCTION:SHA</td>
<td></td>
</tr>
<tr>
<td>SOUR:USER:REG</td>
<td></td>
</tr>
<tr>
<td>SOUR:USER:REP</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LEV:AU</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LEV:IMMed:AMPlitude</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LEV:IMMed:AMPlitude:OFF</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LEV:REF:CHAnal</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LEV:REF:TOLERance</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LIMIT:AMPlitude</td>
<td></td>
</tr>
<tr>
<td>SOUR:VOLT:LIMIT:INPut</td>
<td></td>
</tr>
<tr>
<td>STATus:DEV:COND</td>
<td></td>
</tr>
<tr>
<td>STATus:DEV:ENAB</td>
<td></td>
</tr>
<tr>
<td>STATus:DEV:EVEN</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL )</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>STATus:DEVice:NTRansition</td>
<td>FFT OCT ORD SINE CORR HIST</td>
</tr>
<tr>
<td>STATus:DEVice:PTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:OPERation:CONDition?</td>
<td></td>
</tr>
<tr>
<td>STATus:OPERation:ENABLE</td>
<td></td>
</tr>
<tr>
<td>STATus:OPERation:EVENt?</td>
<td></td>
</tr>
<tr>
<td>STATus:OPERation:NTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:OPERation:PTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:PRESet</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:CONDition?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:ENABLE</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:EVENt?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:LIMit:CONDition?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:LIMit:ENABLE</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:LIMit:EVENt?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:LIMit:NTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:LIMit:PTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:NTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:PTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:VOLTage:CONDition?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:VOLTage:ENABLE</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:VOLTage:EVENt?</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:VOLTage:NTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:QUESTIONable:VOLTage:PTRansition</td>
<td></td>
</tr>
<tr>
<td>STATus:USER:ENABLE</td>
<td></td>
</tr>
<tr>
<td>STATus:USER:EVENt?</td>
<td></td>
</tr>
<tr>
<td>STATus:USER:PULSE?</td>
<td></td>
</tr>
<tr>
<td>SYSTem:BEEPer:IMMediate</td>
<td></td>
</tr>
<tr>
<td>SYSTem:BEEPer:STATe</td>
<td></td>
</tr>
<tr>
<td>SYSTem:COMMunicate:GPIB:SELF:ADDRess</td>
<td></td>
</tr>
<tr>
<td>SYST:COMMunicate:SERial:RECeive:BAUD</td>
<td></td>
</tr>
<tr>
<td>SYST:COMMunicate:SERial:RECeive:BITS</td>
<td></td>
</tr>
<tr>
<td>SYST:COMMunicate:SERial:RECeive:PACE</td>
<td></td>
</tr>
<tr>
<td>SYST:COMM:SERial:RECeive:PARity:CHECK</td>
<td></td>
</tr>
<tr>
<td>SYST:COMM:SERial:RECeive:PARity:TYPE</td>
<td></td>
</tr>
<tr>
<td>SYST:COMM:SERial:RECeive:SBITS</td>
<td></td>
</tr>
<tr>
<td>SYST:COMM:SERial:TRANmit:PACE</td>
<td></td>
</tr>
<tr>
<td>SYSTem:DATE</td>
<td></td>
</tr>
<tr>
<td>SYSTem:ERRor</td>
<td></td>
</tr>
<tr>
<td>SYSTem:FAN:STATe</td>
<td></td>
</tr>
<tr>
<td>SYSTem:FLOG:CLEar</td>
<td></td>
</tr>
<tr>
<td>SYSTem:KEY</td>
<td></td>
</tr>
<tr>
<td>GPIB Command</td>
<td>Instrument Mode (INST:SEL )</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>SYSTem:KLOCk</td>
<td>FFT</td>
</tr>
<tr>
<td>SYSTem:POWer:SOUce</td>
<td></td>
</tr>
<tr>
<td>SYSTem:POWer:STATe</td>
<td></td>
</tr>
<tr>
<td>SYSTem:PRESet</td>
<td></td>
</tr>
<tr>
<td>SYSTem:SET</td>
<td></td>
</tr>
<tr>
<td>SYSTem:TIME</td>
<td></td>
</tr>
<tr>
<td>SYSTem:VERSion</td>
<td></td>
</tr>
<tr>
<td>TEST:LOG:CLEar</td>
<td></td>
</tr>
<tr>
<td>TEST:LONG</td>
<td></td>
</tr>
<tr>
<td>TEST:LONG:RESult?</td>
<td></td>
</tr>
<tr>
<td>TRACe:DATA</td>
<td></td>
</tr>
<tr>
<td>TRACe:WATerfall:DATA</td>
<td></td>
</tr>
<tr>
<td>TRACe:X:DATA</td>
<td></td>
</tr>
<tr>
<td>TRACe:X:UNIT</td>
<td></td>
</tr>
<tr>
<td>TRACe:Z:DATA</td>
<td></td>
</tr>
<tr>
<td>TRACe:Z:UNIT</td>
<td></td>
</tr>
<tr>
<td>TRIGger:EXTernal:FILTER:LPASs:STATe</td>
<td></td>
</tr>
<tr>
<td>TRIGger:EXTernal:LEVel</td>
<td></td>
</tr>
<tr>
<td>TRIGger:EXTernal:RANGe</td>
<td></td>
</tr>
<tr>
<td>TRIGger:IMMediate</td>
<td></td>
</tr>
<tr>
<td>TRIGger:LEVEL</td>
<td></td>
</tr>
<tr>
<td>TRIGger:SLOPe</td>
<td></td>
</tr>
<tr>
<td>TRIGger:SOURce</td>
<td></td>
</tr>
<tr>
<td>TRIGger:STARt</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:HOLDoff</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:LEVel</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:PCOunt</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:RANGe</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:RPM?</td>
<td></td>
</tr>
<tr>
<td>TRIGger:TACHometer:SLOPe</td>
<td></td>
</tr>
</tbody>
</table>
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:DBReference commands for more information.
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.

<table>
<thead>
<tr>
<th>Type of Measurement</th>
<th>Measurement Data Specified Trace Coordinates</th>
<th>Linear or Logarithmic Magnitude</th>
<th>dB Magnitude</th>
<th>Phase/Unwrapped Phase</th>
<th>Real/Imaginary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto correlation</td>
<td>VPK2</td>
<td>DBV</td>
<td>DEG</td>
<td>VPK2</td>
<td>VPK2</td>
</tr>
<tr>
<td>capture buffer</td>
<td>VPK</td>
<td>DBV</td>
<td>DEG</td>
<td>VPK</td>
<td>VPK</td>
</tr>
<tr>
<td>Coherence</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>composite power</td>
<td>VRMS</td>
<td>DBVRMS</td>
<td>DEG</td>
<td>VRMS</td>
<td>VRMS</td>
</tr>
<tr>
<td>cross correlation</td>
<td>VPK2</td>
<td>DBV</td>
<td>DEG</td>
<td>VPK2</td>
<td>VPK2</td>
</tr>
<tr>
<td>cross spectrum</td>
<td>VRMS2</td>
<td>DBVRMS</td>
<td>DEG</td>
<td>VRMS2</td>
<td>VRMS2</td>
</tr>
<tr>
<td>cumulative density function</td>
<td>none</td>
<td>DB</td>
<td>DEG</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>frequency response</td>
<td>none</td>
<td>DB</td>
<td>DEG</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Histogram</td>
<td>COUNT</td>
<td>DBCOUNT</td>
<td>DEG</td>
<td>COUNT</td>
<td></td>
</tr>
<tr>
<td>linear spectrum</td>
<td>VRMS</td>
<td>DBVRMS</td>
<td>DEG</td>
<td>VRMS</td>
<td>VRMS</td>
</tr>
<tr>
<td>normalized variance</td>
<td>none</td>
<td>DB</td>
<td>DEG</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Orbits</td>
<td>not valid</td>
<td>not valid</td>
<td>not valid</td>
<td>VPK</td>
<td></td>
</tr>
<tr>
<td>order track</td>
<td>VRMS</td>
<td>DBVRMS</td>
<td>DEG</td>
<td>VRMS</td>
<td>VRMS</td>
</tr>
<tr>
<td>power spectrum</td>
<td>VRMS2</td>
<td>DBVRMS</td>
<td>DEG</td>
<td>VRMS2</td>
<td>VRMS2</td>
</tr>
<tr>
<td>probability density function</td>
<td>1/V</td>
<td>DB1/V</td>
<td>DEG</td>
<td>1/V</td>
<td></td>
</tr>
<tr>
<td>rpm profile</td>
<td>COUNT</td>
<td>DBCOUNT</td>
<td>DEG</td>
<td>COUNT</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>VPK</td>
<td>DBV</td>
<td>DEG</td>
<td>VPK</td>
<td></td>
</tr>
</tbody>
</table>
### Measurement Data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>windowed time CALC:FEED 'XTIM:VOLT:WIND'</td>
<td>VPK</td>
<td>DBV</td>
<td>DEG</td>
<td>VPK</td>
</tr>
</tbody>
</table>

* 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).
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**

<table>
<thead>
<tr>
<th>Measurement Data</th>
<th>CALC:FEED command (INST:SEL command)</th>
<th>CALC:UNIT:VOLT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Composite Power</td>
<td>CALC:FEED ‘XFR:POW:COMP’ (INST:SEL ORD)</td>
<td>X</td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>CALC:FEED ‘XFR:POW:LIN’ (INST:SEL FFT)</td>
<td>X</td>
</tr>
<tr>
<td>Linear Spectrum</td>
<td>CALC:FEED ‘XFR:POW:LIN’ (INST:SEL SINE)</td>
<td>X</td>
</tr>
<tr>
<td>Order Track</td>
<td>CALC:FEED ‘XORD:TRACK’ (INST:SEL ORD)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED ‘XFR:POW’ (INST:SEL FFT)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED ‘XFR:POW’ (INST:SEL OCT)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED ‘XFR:POW’ (INST:SEL ORD)</td>
<td>X</td>
</tr>
<tr>
<td>Power Spectrum</td>
<td>CALC:FEED ‘XFR:POW’ (INST:SEL ORD)</td>
<td>X</td>
</tr>
</tbody>
</table>

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                       |
## Determining Units

### Table E-6. Y-axis Units when CALC:UNIT:AMPL PEAK and VOLT:RANG:UNIT:USER ON.

<table>
<thead>
<tr>
<th>CALC:UNIT:VOLT</th>
<th>Trace Coordinate Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CALC:FORM [MLIN</td>
</tr>
<tr>
<td>V</td>
<td>EU</td>
</tr>
<tr>
<td>V2</td>
<td>EU2</td>
</tr>
<tr>
<td>V/RTHZ</td>
<td>EU/RTHZ</td>
</tr>
<tr>
<td>V2/HZ</td>
<td>EU2/HZ</td>
</tr>
<tr>
<td>V2S/HZ</td>
<td>EU2S/HZ</td>
</tr>
</tbody>
</table>

### Table E-7. Y-axis Units when CALC:UNIT:AMPL PP and VOLT:RANG:UNIT:USER ON.

<table>
<thead>
<tr>
<th>CALC:UNIT:VOLT</th>
<th>Trace Coordinate Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CALC:FORM [MLIN</td>
</tr>
<tr>
<td>V</td>
<td>EUPP</td>
</tr>
<tr>
<td>V2</td>
<td>EUPP2</td>
</tr>
<tr>
<td>V/RTHZ</td>
<td>EUPP/RTHZ</td>
</tr>
<tr>
<td>V2/HZ</td>
<td>EUPP2/HZ</td>
</tr>
<tr>
<td>V2S/HZ</td>
<td>EUPP2S/HZ</td>
</tr>
</tbody>
</table>

### Table E-8. Y-axis Units when CALC:UNIT:AMPL RMS and VOLT:RANG:UNIT:USER ON.

<table>
<thead>
<tr>
<th>CALC:UNIT:VOLT</th>
<th>Trace Coordinate Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CALC:FORM [MLIN</td>
</tr>
<tr>
<td>V</td>
<td>EURMS</td>
</tr>
<tr>
<td>V2</td>
<td>EURMS2</td>
</tr>
<tr>
<td>V/RTHZ</td>
<td>EURMS/RTHZ</td>
</tr>
<tr>
<td>V2/HZ</td>
<td>EURMS2/HZ</td>
</tr>
<tr>
<td>V2S/HZ</td>
<td>EURMS2S/HZ</td>
</tr>
</tbody>
</table>
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.
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.
SELECT (Res)
CASE 100
Max_i=255
CASE 200
Max_i=511
CASE 400
Max_i=1023
CASE 800
Max_i=2047
CASE ELSE
PRINT "ILLEGAL MEASUREMENT RESOLUTION"
PAUSE
END SELECT
!
REDIM D(Max_i)
!
Dt=Res/Span/(Max_i+1)
FOR T=0 TO 1
FOR I=0 TO Max_i
D(I)=D(I)+SIN(2*PI*Tones(T)*I*Dt)
NEXT I
NEXT T
!
IF Device=800 THEN
OUTPUT @Analyzer;"DISP:FORM QUAD; PROG LOW"
ELSE
OUTPUT @Analyzer;"DISP:FORM ULOW;"
END IF
!
FOR I=0 TO 9
PRINT D(I)
NEXT I
!
OUTPUT @Analyzer;"FORM:DATA REAL, 64"
!
OUTPUT @Analyzer;"CALC1:FEED 'XTIM:VOLT 1';*WAI"
!
OUTPUT @Analyzer;"TRAC:DATA D1,TRAC1;*WAI"
!
OUTPUT @Analyzer;"TRAC:DATA D1,#0;"
OUTPUT @Analyzer_bin;D(*)
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. \( \text{SIN}(2\pi \cdot \text{Tones(T)} \cdot I \cdot D5) \) 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 \text{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 \text{TRAC:DATA} command.

890  \text{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.
Example Programs
ARBSRC

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
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/I BASIC 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 !
80 ON ERROR GOTO Not_ibasic
90 Device=800
100 CLEAR Device
110 GOTO Start_prog
120 !
130 !
140 Not_ibasic:
141 Device=711
150 !
160 Start_prog:
170 OFF ERROR
180 !
190 !
200 ASSIGN @Analyzer TO Device
210 !
220 OUTPUT @Analyzer;"*RST"
230 OUTPUT @Analyzer;"ABORT,:INIT;:WAI"
240 !
250 OUTPUT @Analyzer;"TRAC D1, TRAC1"
260 !
270 !
280 !
290 OUTPUT @Analyzer;"DISP:FORM ULOW"
291 !
292 !
300 OUTPUT @Analyzer;"CALC2: FEED 'D1'; MATH: STAT OFF; :WAI"
310 !
320 END

Example Programs
COPYDATA
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;"CH2: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).
Example Programs

FREQRESP

480!
490 OUTPUT @Analyzer; "CALC1: FEED ‘XFR: POW: RAT 2, 1’; * WA1"
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_be; 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
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 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 !
210 DIM A(400)
220 OUTPUT @Analyzer;"*RST"
230 OUTPUT @Analyzer;"ABORT;:INIT; *WAI"
240 OUTPUT @Analyzer;":INIT:CONT OFF"
250 !
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 ENTER @Analyzer USING "%K"; A(I)
420 IF I THEN PRINT "Data point ":I, A(I)
430 NEXT I
440 !
450 !
460 END
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 !
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 ) &"; 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$("SYSTEM ID"),"Agilent 35670") THEN
160 Address=800
170 ELSE
180 Address=711
190 END IF
200 !
210 ASSIGN @Analyzer TO Address
220 ASSIGN @Analyzer_bin TO Address;FORMAT OFF
230 !
240 !
250 OUTPUT @Analyzer;"SYST:PRES"
260 OUTPUT @Analyzer;"VOLT1:RANG 2 VPK"
270 OUTPUT @Analyzer;"OUTP ON"
280 OUTPUT @Analyzer;"SOUR:VOLT 2 VPK"
290 OUTPUT @Analyzer;"SOUR:FREQ 100 HZ"
300 OUTPUT @Analyzer;"CALC:FEED 'TCAP 1'"
310 OUTPUT @Analyzer;"DISP:WIND1:TRAC:Y:TOP 5; PDIV 1"
320 OUTPUT @Analyzer;"TCAP"
330 !
340 OUTPUT @Analyzer;"*OPC?"
350 ENTER @Analyzer;Opc
360 !
370 !
380 OUTPUT @Analyzer;"FORM REAL,64"
390 !
400 OUTPUT @Analyzer;"SENSE:DATA? TCAP1"
410 ENTER @Analyzer USING "%,A,D";Resp$
420 ENTER @Analyzer USING "%,"&VAL$(l)"D";Bytes
430 !
440 Points=Bytes/8
450 Records=Points/1024.
460 !

Program listing continues on page F-23
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
62 ! Curve Fit table.
64 ! This program will also distinguish between IBASIC and BASIC.
66 !---------------------------------------------------------------
68 !
70 ! ON_ERROR GOTO Not_ibasic
72 !
74 Device = 800
76 CLEAR Device
78 GOTO Start_prog
80 !
82 ! Not_ibasic:
84 Device=711
86 Start_prog:
88 OFF ERROR
90 !
92 ASSIGN @Analyzer TO Device !
94 DIM A(5000)
96 ASSIGN @Analyzer_bin TO Device; FORMAT OFF
98 OUTPUT @Analyzer;"FORM: DATA REAL,64"
100 !
102 OUTPUT @Analyzer;"CALC1: CFIT: DATA?"
104 !
106 ENTER @Analyzer USING "%A,D"; A$, Digits
108 ENTER @Analyzer USING "%" & VAL$(Digits) & "D"; Num_of_bytes
110 Num_points=Num_of_bytes DIV 8
112 !
114 REDIM A(Num_points-1)
116 ENTER @Analyzer_bin; A(*)
118 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.
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 "CURVE FIT TERMS"
690 PRINT "POLES ZERO'S"
700 PRINT ""
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 ! BASIC/IBASIC Programs; Octave Waterfall Dump
30 !
40 ! this program is divided into three parts. (1) set up the source, (2) set up
50 ! and take a MEASUREMENT, (3) dump the data out of the instrument and print
60 ! out the information. The information is in dBSPLrms units
70 !
80 ! Some items of note.
90 !
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
210 ! Modes.
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.
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 -340 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:


<File> path and file name
/U show this usage
/I: <code> Select code (default = 7 for Agilent Technologies card<>, 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 file named CAPT1.TIM on a PC using an 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

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 18-8, 18-11 preview 18-14 repeat 18-14 RMS 18-19 termination control time 18-14 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
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 6-42
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 15-14, 15-30
upper limit lines 15-27

disk operations, format
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
   fast X-axis value 8-32
   number of data points 18-27
   number of orders 18-51
Nyquist diagram 8-24 - 8-25
   overall band 6-30
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 \(18-84\)
labelling \(18-82\)
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) \(F-11 - F-14, F-17 - F-20, F-24 - F-39\)
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
INDEX (Continued)

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
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
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
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 18-26
source for measurement data 12-9
specifying channels 18-4
trigger level 24-6
inserting entries, data tables 6-63
Instrument BASIC 8-17
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 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 6-66
for frequency data 6-67
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, 6-93
    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
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
    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
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
caracter format 21-10
caracter 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 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

definitions 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

1/12 octave band measurement 18-38

1/3 octave band measurement 18-38

1/12 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 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
   initiating 11-2
input channels
   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

xii
unfiltered time record, displaying 6-25
uniform window 18-90
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
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
tvolt units 6-119
volt} {\plain \super \f1 2} {\plain \f1 units 6-119
masking traces 8-43
number of traces 6-128
saving a trace 6-135
selecting a trace 6-132 - 6-133
selecting a slice 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
xder 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
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
*WAI 3-20
<table>
<thead>
<tr>
<th>If you are thinking about...</th>
<th>And you want to...</th>
<th>Then read...</th>
</tr>
</thead>
</table>
| ♦ Unpacking and installing the Agilent 35670A | Install the Agilent 35670A Dynamic Signal Analyzer  
Do operation verification or performance verification tests | Agilent 35670A Installation and Verification Guide  
Agilent 35670A Installation and Verification Guide |
| ♦ Getting started | Make your first measurements with your new analyzer  
Review measurement basics  
Learn what each key does | Agilent 35670A Quick Start Guide  
Agilent 35670A Operator’s Guide  
Use the analyzer’s [Help] key |
| ♦ Making measurements | Learn how to make typical measurements with the Agilent 35670A  
Understand each of the analyzer’s instrument modes | Agilent 35670A Operator’s Guide  
Agilent 35670A Operator’s Guide |
| ♦ Creating automated measurements  
(Instrument Basic is Option 1C2) | Learn the Instrument Basic interface  
Record keystrokes for a particular measurement  
Program with Instrument Basic | Using Instrument Basic with the Agilent 35670A  
Agilent 35670A Quick Start Guide  
Instrument Basic User’s Handbook |
| ♦ Remote operation | Learn about the GPIB  
Learn how to program with GPIB  
Find specific GPIB commands | GPIB Programmer’s Guide  
GPIB Programming with the Agilent 35670A  
Agilent 35670A GPIB Commands: Quick Reference |
| ♦ Using analyzer data with a PC application | Display or plot analyzer data on or from a Personal Computer  
Transfer analyzer data to a PC software application format  
Transfer data from a PC software application format to the analyzer (for example, to load data into a data register) | Standard Data Format Utilities: User’s Guide |
| ♦ Servicing the analyzer | Adjust, troubleshoot, or repair the analyzer | Agilent 35670A Service Guide |
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:

<table>
<thead>
<tr>
<th>United States</th>
<th>Test and Measurement Call Center</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(800) 452-4844 (Toll free in US)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Canada</th>
<th>(905) 206-4725</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Europe</th>
<th>(31 20) 547 9900</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Japan</th>
<th>Measurement Assistance Center</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(81) 426 56 7832</td>
</tr>
<tr>
<td></td>
<td>(81) 426 56 7840 (FAX)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latin America</th>
<th>(305) 267 4245</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(305) 267 4288 (FAX)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Australia/New Zealand</th>
<th>1 800 629 486 (Australia)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0800 738 378 (New Zealand)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Asia-Pacific</th>
<th>(852) 2589 7777</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(FAX) (852) 2506 9285</td>
</tr>
</tbody>
</table>