

**Agilent Technologies  
E8257D/67D, & E8663D  
PSG Signal Generators**

**SCPI Command Reference**



**Agilent Technologies**

# Notices

© Agilent Technologies, Inc. 2006-2011

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

## Manual Part Number

E8251-90356

## Edition

December 2011

Printed in USA

Agilent Technologies, Inc.  
3501 Stevens Creek Blvd.  
Santa Clara, CA 95052 USA

## Warranty

**The material contained in this manual is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this manual that conflict with these terms, the warranty terms in the separate agreement shall control.**

## Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

## Restricted Rights Legend

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in

Commercial Computer Software or Computer Software Documentation).

## Safety Notices

### CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

### WARNING

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

**1 Using this Guide**

How the SCPI Information is Organized . . . . .	1
SCPI Listings . . . . .	1
Subsystem Groupings by Chapter . . . . .	1
Front Panel Operation Cross Reference . . . . .	1
Supported Models and Options per Command . . . . .	2
SCPI Basics . . . . .	2
Common Terms . . . . .	2
Command Syntax . . . . .	2
Command Types . . . . .	4
Command Tree . . . . .	4
Command Parameters and Responses . . . . .	5
Program Messages . . . . .	9
File Name Variables . . . . .	10
ARB Waveform File Directories . . . . .	11
MSUS (Mass Storage Unit Specifier) Variable . . . . .	11
Quote Usage with SCPI Commands . . . . .	12
Binary, Decimal, Hexadecimal, and Octal Formats . . . . .	13

**2 System Commands**

Calibration Subsystem (:CALibration) . . . . .	16
:DCFM . . . . .	16
:IQ . . . . .	16
:IQ:DC . . . . .	16
:IQ:DEFault . . . . .	17
:IQ:FULL . . . . .	17
:IQ:START . . . . .	18
:IQ:STOP . . . . .	18
:WBIQ . . . . .	18
:WBIQ:DC . . . . .	19
:WBIQ:DEFault . . . . .	19
:WBIQ:FULL . . . . .	20
:WBIQ:START . . . . .	20
:WBIQ:STOP . . . . .	20
Communication Subsystem (:SYSTem:COMMunicate) . . . . .	21
:GPIB:ADDRes . . . . .	21
:GTLocal . . . . .	21
:LAN:CONFig . . . . .	22
:LAN:GATEway . . . . .	22
:LAN:HOSTName . . . . .	22
:LAN:IP . . . . .	23

---

# Contents

:LAN:SUBNet. . . . .	23
:PMEter:ADDRESS. . . . .	24
:PMEter:CHANnel . . . . .	24
:PMEter:IDN . . . . .	25
:PMEter:TIMEout . . . . .	25
:SERial:BAUD . . . . .	26
:SERial:ECHO . . . . .	26
:SERial:RESet . . . . .	26
:SERial:TOUT. . . . .	27
Diagnostic Subsystem (:DIAGnostic[:CPU]:INFORMation) . . . . .	28
:BOARds. . . . .	28
:CCOunt:ATTenuator . . . . .	28
:CCOunt:PON. . . . .	28
:DISPlay:OTIME . . . . .	28
:LICENse:AUXiliary. . . . .	28
:OPTions. . . . .	29
:OPTions:DETail . . . . .	29
:OTIME . . . . .	29
:REVision . . . . .	29
:SDATe. . . . .	29
Display Subsystem (:DISPlay) . . . . .	30
:ANNOtation:AMPLitude:UNIT . . . . .	30
:ANNOtation:CLOCK:DATE:FORMat . . . . .	30
:ANNOtation:CLOCK[:STATe] . . . . .	31
:BRIGHtness . . . . .	31
:CAPTure . . . . .	32
:CONTrast. . . . .	32
:INVerse . . . . .	32
:REMote . . . . .	33
[:WINDow][:STATe]. . . . .	33
IEEE 488.2 Common Commands. . . . .	34
*CLS . . . . .	34
*ESE . . . . .	34
*ESE? . . . . .	34
*ESR? . . . . .	35
*IDN?. . . . .	35
*OPC . . . . .	35
*OPC? . . . . .	35
*PSC . . . . .	36
*PSC? . . . . .	36
*RCL . . . . .	36

*RST . . . . .	36
*SAV . . . . .	37
*SRE . . . . .	37
*SRE?. . . . .	37
*STB?. . . . .	38
*TRG . . . . .	38
*TST?. . . . .	38
*WAI . . . . .	38
Low-Band Filter Subsystem . . . . .	39
[:SOURce]:LBFilter . . . . .	39
Memory Subsystem (:MEMory) . . . . .	39
:CATalog:BINary . . . . .	39
:CATalog:BIT . . . . .	39
:CATalog:DMOD . . . . .	40
:CATalog:FIR . . . . .	40
:CATalog:FSK . . . . .	40
:CATalog:IQ . . . . .	41
:CATalog:LIST . . . . .	41
:CATalog:MMod. . . . .	41
:CATalog:MTONe . . . . .	42
:CATalog:SEQ. . . . .	42
:CATalog:SHApe . . . . .	42
:CATalog:STATe . . . . .	43
:CATalog:UFLT . . . . .	43
:CATalog[:ALL]. . . . .	43
:COPY[:NAME] . . . . .	44
:DATA . . . . .	44
:DATA:APPend . . . . .	45
:DATA:BIT . . . . .	46
:DATA:FIR . . . . .	47
:DATA:FSK . . . . .	47
:DATA:IQ. . . . .	49
:DATA:PRAM:FILE:BLOCK. . . . .	50
:DATA:PRAM:FILE:LIST . . . . .	51
:DATA:PRAM? . . . . .	51
:DATA:PRAM:BLOCK . . . . .	52
:DATA:PRAM:LIST. . . . .	52
:DATA:SHApe. . . . .	52
:DATA:UNPRotected. . . . .	53
:DELete:ALL . . . . .	54
:DELete:BINary . . . . .	54

---

# Contents

:DELeTe:BIT. . . . .	54
:DELeTe:DMOD. . . . .	55
:DELeTe:FIR. . . . .	55
:DELeTe:FSK . . . . .	55
:DELeTe:IQ . . . . .	55
:DELeTe:LIST . . . . .	55
:DELeTe:MDMod . . . . .	56
:DELeTe:MTONe . . . . .	56
:DELeTe:SEQ . . . . .	56
:DELeTe:SHAPE . . . . .	56
:DELeTe:STATE. . . . .	56
:DELeTe:UFLT . . . . .	57
:DELeTe[:NAME]. . . . .	57
:FREE[:ALL] . . . . .	57
:LOAD:LIST. . . . .	57
:MOVE . . . . .	58
:STATe:COMMeNt . . . . .	58
:STORe:LIST . . . . .	58
Mass Memory Subsystem (:MMEMory) . . . . .	59
:CATalog. . . . .	59
:COPY . . . . .	60
:DATA . . . . .	60
:DELeTe:NVWFm. . . . .	61
:DELeTe:WFM. . . . .	61
:DELeTe[:NAME]. . . . .	61
:HEADer:CLEAr . . . . .	62
:HEADer:DESCRiption . . . . .	62
:HEADer:ID? . . . . .	63
:LOAD:LIST. . . . .	63
:MOVE . . . . .	63
:STORe:LIST . . . . .	63
Output Subsystem (:OUTPut) . . . . .	65
:BLANking:AUTO . . . . .	65
:BLANking:[STATe] . . . . .	65
:MODulation[:STATe]. . . . .	66
:SETTled? . . . . .	66
:SETTled:POLarity . . . . .	66
:SETTled:RETRace . . . . .	67
:SETTled:RFOff . . . . .	67
[:STATe]. . . . .	68
Route Subsystem (:ROUTe:HARDWare:DGENerator). . . . .	69

:INPut:BPOLarity . . . . .	69
:INPut:CPOLarity . . . . .	69
:INPut:DPOLarity . . . . .	69
:INPut:SPOLarity . . . . .	70
:IPOLarity:BGATE . . . . .	70
:IPOLarity:CLOCK . . . . .	71
:IPOLarity:DATA . . . . .	71
:IPOLarity:SSYNc . . . . .	71
:OPOLarity:CLOCK. . . . .	72
:OPOLarity:DATA . . . . .	72
:OPOLarity:EVENT[1] 2 3 4. . . . .	73
:OPOLarity:SSYNc . . . . .	73
:OUTPut:CPOLarity . . . . .	73
:OUTPut:DCS[:STATE] . . . . .	74
:OUTPut:DPOLarity . . . . .	74
:OUTPut:EPOL[1] 2 3 4 . . . . .	75
:OUTPut:SPOLarity . . . . .	75
Status Subsystem (:STATus) . . . . .	76
:OPERation:BASEband:CONDition . . . . .	76
:OPERation:BASEband:ENABle . . . . .	76
:OPERation:BASEband:NTRansition . . . . .	76
:OPERation:BASEband:PTRansition . . . . .	77
:OPERation:BASEband[:EVENT] . . . . .	77
:OPERation:CONDition . . . . .	78
:OPERation:ENABle . . . . .	78
:OPERation:NTRansition . . . . .	78
:OPERation:PTRansition . . . . .	79
:OPERation[:EVENT] . . . . .	79
:PRESet . . . . .	80
:QUEStionable:CALibration:CONDition . . . . .	80
:QUEStionable:CALibration:ENABle . . . . .	80
:QUEStionable:CALibration:NTRansition . . . . .	81
:QUEStionable:CALibration:PTRansition . . . . .	81
:QUEStionable:CALibration[:EVENT] . . . . .	82
:QUEStionable:CONDition . . . . .	82
:QUEStionable:ENABle . . . . .	82
:QUEStionable:FREQuency:CONDition . . . . .	83
:QUEStionable:FREQuency:ENABle . . . . .	83
:QUEStionable:FREQuency:NTRansition . . . . .	83
:QUEStionable:FREQuency:PTRansition . . . . .	84
:QUEStionable:FREQuency[:EVENT]. . . . .	84

---

# Contents

:QUEStionable:MODulation:CONDition . . . . .	85
:QUEStionable:MODulation:ENABle . . . . .	85
:QUEStionable:MODulation:NTRansition . . . . .	85
:QUEStionable:MODulation:PTRansition . . . . .	86
:QUEStionable:MODulation[:EVENT] . . . . .	86
:QUEStionable:NTRansition . . . . .	86
:QUEStionable:POWER:CONDition . . . . .	87
:QUEStionable:POWER:ENABle . . . . .	87
:QUEStionable:POWER:NTRansition . . . . .	88
:QUEStionable:POWER:PTRansition . . . . .	88
:QUEStionable:POWER[:EVENT]. . . . .	88
:QUEStionable:PTRansition . . . . .	89
:QUEStionable[:EVENT] . . . . .	89
System Subsystem (:SYStem) . . . . .	90
:ALTErnate . . . . .	90
:ALTErnate:STAtE . . . . .	90
:CAPability . . . . .	90
:DATE . . . . .	91
:ERRor[:NEXT]. . . . .	91
:ERRor:SCPI[:SYNTax] . . . . .	91
:FILEsystem:SAFEmode . . . . .	92
:HELP:MODE . . . . .	92
:IDN . . . . .	92
:LANGuage . . . . .	93
:OEMHead:FREQuency:START . . . . .	94
:OEMHead:FREQuency:STOP . . . . .	94
:OEMHead:SELEct. . . . .	94
:OEMHead:FREQuency:BAND WR15 WR12 WR10 WR8 WR6 WR5 WR3 . . . . .	95
:OEMHead:FREQuency:MULTiplier . . . . .	96
:PON:TYPE . . . . .	96
:PRESet . . . . .	97
:PRESet:ALL . . . . .	97
:PRESet:LANGuage . . . . .	97
:PRESet:PERsistent. . . . .	98
:PRESet:PN9 . . . . .	98
:PRESet:TYPE . . . . .	99
:PRESet[:USER]:SAVE . . . . .	99
:SECurity:DISPlay . . . . .	99
:SECurity:ERASeall . . . . .	100
:SECurity:LEVel . . . . .	100
:SECurity:LEVel:STAtE. . . . .	101



:SECurity:OVERwrite . . . . .	102
:SECurity:SANitize . . . . .	102
:SSAVer:DELay . . . . .	102
:SSAVer:MODE . . . . .	103
:SSAVer:STAtE . . . . .	103
:TIME. . . . .	103
:VERSion. . . . .	104
Trigger Subsystem . . . . .	105
:ABORt. . . . .	105
:INITiate:CONTinuous[:ALL] . . . . .	105
:INITiate[:IMMediate][:ALL]. . . . .	105
:TRIGger:OUTPut:POLarity . . . . .	106
:TRIGger[:SEQuence]:SLOPe . . . . .	106
:TRIGger[:SEQuence]:SOURce . . . . .	106
:TRIGger[:SEQuence][:IMMediate]. . . . .	107
Unit Subsystem (:UNIT). . . . .	108
:POWer . . . . .	108

### **3 Basic Function Commands**

Correction Subsystem ([:SOURce]:CORRection). . . . .	110
:FLATness:LOAD . . . . .	110
:FLATness:PAIR . . . . .	110
:FLATness:POINts . . . . .	111
:FLATness:PRESet. . . . .	111
:FLATness:STORe . . . . .	111
[:STAtE]. . . . .	112
Frequency Subsystem ([:SOURce]). . . . .	113
:FREQuency:CENTer . . . . .	113
:FREQuency:CHANnels:BAND . . . . .	114
:FREQuency:CHANnels:NUMBer . . . . .	116
:FREQuency:CHANnels[:STAtE] . . . . .	117
:FREQuency:FIXed . . . . .	117
:FREQuency:MANual . . . . .	118
:FREQuency:MODE . . . . .	119
:FREQuency:MULTIplier . . . . .	119
:FREQuency:OFFSet. . . . .	120
:FREQuency:OFFSet:STAtE . . . . .	120
:FREQuency:REFerence . . . . .	121
:FREQuency:REFerence:SET . . . . .	121
:FREQuency:REFerence:STAtE . . . . .	121
:FREQuency:SPAN . . . . .	122

---

# Contents

:FREQUENCY:START	122
:FREQUENCY:STOP	123
:FREQUENCY[:CW]	124
:FREQUENCY[:CW]:STEP[:INCREMENT]	125
:FREQUENCY:SYNTHESIS	125
:PHASE:REFERENCE	125
:PHASE[:ADJUST]	125
:ROSCILLATOR:BANDWIDTH:DEFAULTS	126
:ROSCILLATOR:BANDWIDTH:EXTERNAL	126
:ROSCILLATOR:BANDWIDTH:INTERNAL	126
:ROSCILLATOR:SOURCE:AUTO	127
:ROSCILLATOR:SOURCE	127
List/Sweep Subsystem ([:SOURCE])	128
:LIST:DIRECTION	128
:LIST:DWELL	129
:LIST:DWELL:POINTS	129
:LIST:DWELL:TYPE	129
:LIST:FREQUENCY	130
:LIST:FREQUENCY:POINTS	131
:LIST:MANUAL	131
:LIST:MODE	131
:LIST:POWER	132
:LIST:POWER:POINTS	132
:LIST:RETRACE	132
:LIST:TRIGGER:SOURCE	133
:LIST:TYPE	133
:LIST:TYPE:LIST:INITIALIZE:FSTEP	134
:LIST:TYPE:LIST:INITIALIZE:PRESET	134
:SWEPT:CONTROL:STATE	134
:SWEPT:CONTROL:TYPE	135
:SWEPT:DWELL	136
:SWEPT:GENERATION	136
:SWEPT:MODE	137
:SWEPT:POINTS	137
:SWEPT:TIME	138
:SWEPT:TIME:AUTO	138
Marker Subsystem–Option 007 ([:SOURCE])	139
:MARKER:AMPLITUDE[:STATE]	139
:MARKER:AMPLITUDE:VALUE	139
:MARKER:AOFF	139
:MARKER:DELTA?	140

:MARKer[0,1,2,3,4,5,6,7,8,9]:FREQUency	140
:MARKer:MODE	140
:MARKer:REFerence	141
:MARKer[0,1,2,3,4,5,6,7,8,9][:STATe]	141
Power Subsystem ([:SOURce]:POWER)	142
:ALC:BANDwidth BWIDTH	142
:ALC:BANDwidth BWIDTH:AUTO	142
:ALC:LEVel	143
:ALC:SEARch	143
:ALC:SEARch:REFerence	144
:ALC:SEARch:REFerence:LEVel	145
:ALC:SEARch:SPAN:START	145
:ALC:SEARch:SPAN:STOP	145
:ALC:SEARch:SPAN:TYPE FULL USER	146
:ALC:SEARch:SPAN[:STATe] ON OFF 1 0	146
:ALC:SOURce	146
:ALC:SOURce:EXTernal:COUPling	147
:ALC[:STATe]	147
:ATTenuation	147
:ATTenuation:AUTO	148
:MODE	148
:NOISe[:STATe]	149
:POWER:LIMit[:MAX]:ADJust	150
:POWER:LIMit[:MAX]	150
:PROTection:STATe	150
:REFerence	151
:REFerence:STATe	151
:START	152
:STOP	152
[:LEVel][:IMMediate]:OFFSet	153
[:LEVel][:IMMediate][:AMPLitude]	153
[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement]	154
Trigger Sweep Subsystem ([:SOURce])	155
:TSWEEP	155

## 4 Analog Commands

Amplitude Subsystem ([:SOURce])	157
:AM[1] 2....	157
:AM:INTernal:FREQUency:STEP[:INCRement]	158
:AM:MODE	158
:AM:POLarity	159

---

## Contents

:AM:WIDeband:SENSitivity	159
:AM:WIDeband:STATe	159
:AM[1] 2:EXTernal[1] 2:COUPLing	160
:AM[1] 2:EXTernal[1] 2:IMPedance	160
:AM[1] 2:INTernal[1] 2:FREQuency	161
:AM[1] 2:INTernal[1]:FREQuency:ALTErnate	161
:AM[1] 2:INTernal[1]:FREQuency:ALTErnate:AMPLitude:PERCent	162
:AM[1] 2:INTernal[1] 2:FUNCTion:NOISE	162
:AM[1] 2:INTernal[1] 2:FUNCTion:RAMP	162
:AM[1] 2:INTernal[1] 2:FUNCTion:SHAPE	163
:AM[1] 2:INTernal[1]:SWEep:RATE	163
:AM[1] 2:INTernal[1]:SWEep:TRIGger	163
:AM[1] 2:SOURce	164
:AM[1] 2:STATe	164
:AM[1] 2:TYPE	165
:AM[1] 2[:DEPTH]:EXPOntial	165
:AM[1] 2[:DEPTH][:LINEar]	166
:AM[1] 2[:DEPTH][:LINEar]:TRACk	166
:AM[:DEPTH]:STEP[:INCRement]	167
Frequency Modulation Subsystem ([:SOURce])	168
:FM[1] 2....	168
:FM:INTernal:FREQuency:STEP[:INCRement]	169
:FM[1] 2:EXTernal[1] 2:COUPLing	169
:FM[1] 2:EXTernal[1] 2:IMPedance	169
:FM[1] 2:INTernal[1]:FREQuency:ALTErnate	170
:FM[1] 2:INTernal[1]:FREQuency:ALTErnate:AMPLitude:PERCent	170
:FM[1] 2:INTernal[1]:SWEep:RATE	171
:FM[1] 2:INTernal[1]:SWEep:TRIGger	171
:FM[1] 2:INTernal[1] 2:FREQuency	172
:FM[1] 2:INTernal[1] 2:FUNCTion:NOISE	172
:FM[1] 2:INTernal[1] 2:FUNCTion:RAMP	173
:FM[1] 2:INTernal[1] 2:FUNCTion:SHAPE	173
:FM[1] 2:SOURce	173
:FM[1] 2:STATe	174
:FM[1] 2[:DEViation]	174
:FM[1] 2[:DEViation]:TRACk	175
Low Frequency Output Subsystem ([:SOURce]:LFOOutput)	176
:LFOOutput:AMPLitude	176
:LFOOutput:FUNCTion[1] 2:FREQuency	176
:LFOOutput:FUNCTion[1]:FREQuency:ALTErnate	177
:LFOOutput:FUNCTion[1]:FREQuency:ALTErnate:AMPLitude:PERCent	177

:LFOutput:FUNCTION[1] 2:SHAPE . . . . .	177
:LFOutput:FUNCTION:[1] 2:SHAPE:NOISE. . . . .	178
:LFOutput:FUNCTION[1] 2:SHAPE:RAMP . . . . .	178
:LFOutput:FUNCTION[1]:SWEep:RATE . . . . .	179
:FUNCTION[1]:SWEep:TRIGger . . . . .	179
:LFOutput:SOURce . . . . .	180
:LFOutput:STATe . . . . .	180
Phase Modulation Subsystem ([:SOURce]) . . . . .	181
:PM[1] 2: . . . . .	181
:PM:INTernal:FREQuency:STEP[:INCRement]. . . . .	181
:PM[1] 2:BANDwidth BWIDth . . . . .	182
:PM[1] 2:EXTernal[1] 2:COUPLing. . . . .	182
:PM[1] 2:EXTernal[1] 2:IMPedance . . . . .	183
:PM[1] 2:INTernal[1]:FREQuency . . . . .	183
:PM[1] 2:INTernal[1]:FREQuency:ALTerNate . . . . .	183
:PM[1] 2:INTernal[1] 2:FUNCTion:NOISE. . . . .	184
:PM[1] 2:INTernal[1] 2:FUNCTion:RAMP . . . . .	184
:PM[1] 2:INTernal[1]:FREQuency:ALTerNate:AMPLitude:PERCent . . . . .	185
:PM[1] 2:INTernal[1]:FUNCTion:SHAPE. . . . .	185
:PM[1] 2:INTernal2:FUNCTion:SHAPE. . . . .	186
:PM[1] 2:INTernal[1]:SWEep:RATE . . . . .	186
:PM[1] 2:INTernal[1]:SWEep:TRIGger . . . . .	186
:PM[1] 2:SOURce . . . . .	187
:PM[1] 2:STATe . . . . .	187
:PM[1] 2[:DEViation] . . . . .	188
:PM[1] 2[:DEViation]:TRACk . . . . .	189
:PM[:DEViation]:STEP[:INCRement]. . . . .	189
Pulse Modulation Subsystem ([:SOURce]) . . . . .	190
:PULM:EXTernal:POLarity NORMal:INVerted . . . . .	190
:PULM:INTernal[1]:DELay . . . . .	190
:PULM:INTernal[1]:DELay:STEP . . . . .	191
:PULM:INTernal[1]:FREQuency . . . . .	191
:PULM:INTernal[1]:FREQuency:STEP . . . . .	192
:PULM:INTernal[1]:PERiod . . . . .	192
:PULM:INTernal[1]:PERiod:STEP[:INCRement] . . . . .	193
:PULM:INTernal[1]:PWIDTH . . . . .	193
:PULM:INTernal[1]:PWIDTH:STEP . . . . .	194
:PULM:INTernal . . . . .	194
:PULM:SOURce . . . . .	195
:PULM:STATe . . . . .	195

---

# Contents

## 5 Digital Modulation Commands

All Subsystem–Option 601 and 602 ([:SOURce]). . . . .	197
:RADio:ALL:OFF. . . . .	197
AWGN ARB Subsystem–Option 403 ([:SOURce]:RADio:AWGN:ARB) . . . . .	198
:BWIDth. . . . .	198
:IQ:EXtErnal:FILTer . . . . .	198
:IQ:EXtErnal:FILTer:AUTO . . . . .	198
:HEADer:CLEar . . . . .	199
:HEADer:SAVE. . . . .	199
:IQ:MODulation:ATTen . . . . .	199
:IQ:MODulation:ATTen:AUTO. . . . .	199
:IQ:MODulation:FILTer . . . . .	200
:IQ:MODulation:FILTer:AUTO. . . . .	200
:MDEStination:AAMplitude . . . . .	200
:MDEStination:ALCHold . . . . .	201
:MDEStination:PULSe . . . . .	202
:MPOLarity:MARKer1 2 3 4 . . . . .	203
:LENGth . . . . .	203
:REFerence:EXtErnal:FREQUency. . . . .	204
:REFerence[:SOURce] . . . . .	204
:SCLock:RATE . . . . .	204
:SEED . . . . .	205
[:STATe]. . . . .	205
AWGN Real–Time Subsystem–Option 403 ([:SOURce]:RADio:AWGN:RT). . . . .	205
:BWIDth. . . . .	205
[:STATe]. . . . .	205
Custom Subsystem–Option 601 and 602 ([:SOURce]:RADio:CUSTom) . . . . .	206
:ALPha. . . . .	206
:BBCLock . . . . .	206
:BBT . . . . .	207
:BRATe. . . . .	207
:BURSt:SHAPE:FALL:DELay. . . . .	208
:BURSt:SHAPE:FALL:TIME . . . . .	209
:BURSt:SHAPE:FDELay. . . . .	209
:BURSt:SHAPE:FTIME. . . . .	210
:BURSt:SHAPE:RDELay . . . . .	211
:BURSt:SHAPE:RISE:DELay . . . . .	211
:BURSt:SHAPE:RISE:TIME. . . . .	212
:BURSt:SHAPE:RTIME . . . . .	212
:BURSt:SHAPE[:TYPE] . . . . .	213
:CHANnel . . . . .	213

:DACS:ALIGn . . . . .	214
:DATA . . . . .	214
:DATA:EDIGitalbus . . . . .	214
:DATA:FIX4 . . . . .	215
:DATA:PRAM . . . . .	215
:DENCode . . . . .	215
:EDATa:DELay . . . . .	217
:EDCLock . . . . .	217
:EREFerence . . . . .	217
:EREFerence:VALue . . . . .	218
:FILTer . . . . .	218
:IQ:SCALE . . . . .	219
:MODulation:FSK[:DEViation]. . . . .	219
:MODulation:MSK[:PHASe] . . . . .	220
:MODulation:UFSK . . . . .	220
:MODulation:UIQ . . . . .	220
:MODulation[:TYPE] . . . . .	221
:POLarity[:ALL] . . . . .	221
:SRATe . . . . .	222
:STANdard:SELect . . . . .	223
:TRIGger:TYPE . . . . .	224
:TRIGger:TYPE:CONTinuous[:TYPE] . . . . .	225
:TRIGger:TYPE:GATE:ACTive . . . . .	226
:TRIGger[:SOURce] . . . . .	227
:TRIGger[:SOURce]:EXTernal:DELay . . . . .	228
:TRIGger[:SOURce]:EXTernal:DELay:STATe . . . . .	228
:TRIGger[:SOURce]:EXTernal:SLOPe . . . . .	229
:TRIGger[:SOURce]:EXTernal[:SOURce] . . . . .	229
[:STATe] . . . . .	230
:VCO:CLOCK . . . . .	230
Digital Modulation Subsystem ([:SOURce]:DM) . . . . .	231
:EXTernal:Filter . . . . .	231
:EXTernal:Filter:AUTO . . . . .	231
:EXTernal:HCRest . . . . .	232
:EXTernal:POLarity . . . . .	232
:EXTernal:SOURce . . . . .	233
:IQADjustment:DELay . . . . .	233
:IQADjustment:EXTernal:COFFset . . . . .	234
:IQADjustment:EXTernal:DIOFFset . . . . .	234
:IQADjustment:EXTernal:DQOFFset . . . . .	235
:IQADjustment:EXTernal:GAIN . . . . .	235

---

## Contents

:IQADjustment:EXTErnal:IOFFset . . . . .	236
:IQADjustment:EXTErnal:IQATten . . . . .	236
:IQADjustment:EXTErnal:QOFFset . . . . .	237
:IQADjustment:GAIN . . . . .	237
:IQADjustment:IOFFset . . . . .	238
:IQADjustment:QOFFset . . . . .	238
:IQADjustment:QSKew . . . . .	239
:IQADjustment:SKEW . . . . .	239
:IQADjustment:SKEW:Path . . . . .	240
:IQADjustment[:STATe] . . . . .	241
:MODulation:ATTen . . . . .	241
:MODulation:ATTen:AUTO . . . . .	241
:MODulation:ATTen:EXTErnal . . . . .	242
:MODulation:ATTenn:EXTErnal:LEVel . . . . .	242
:MODulation:ATTenn:EXTErnal:LEVel:MEASurement . . . . .	243
:MODulation:ATTen:OPTimize:BANDwidth . . . . .	243
:MODulation:FILTer . . . . .	243
:MODulation:FILTer:AUTO . . . . .	244
:POLarity[:ALL] . . . . .	244
:SKEW:PATH . . . . .	245
:SKEW[:STATe] . . . . .	245
:SOURce . . . . .	246
:SRATio . . . . .	246
:STATe . . . . .	247
Dual ARB Subsystem–Option 601 or 602 ([:SOURce]:RADio:ARB) . . . . .	247
:BASEband:FREQuency:OFFSet . . . . .	247
:CLIPping . . . . .	248
:DACS:ALIGn . . . . .	248
:GENerate:SINE . . . . .	249
:HEADer:CLEar . . . . .	249
:HEADer:RMS . . . . .	250
:HEADer:SAVE . . . . .	251
:IQ:EXTErnal:FILTer . . . . .	251
:IQ:EXTErnal:FILTer:AUTO . . . . .	251
:IQ:MODulation:ATTen . . . . .	252
:IQ:MODulation:ATTen:AUTO . . . . .	252
:IQ:MODulation:FILTer . . . . .	253
:IQ:MODulation:FILTer:AUTO . . . . .	253
:MARKer:CLEar . . . . .	254
:MARKer:CLEar:ALL . . . . .	255
:MARKer:ROTate . . . . .	255



:MARKer:[SET]	256
:MDEStination:AAMPlitude	258
:MDEStination:ALCHold	258
:MDEStination:PULSe	259
:MPOLarity:MARKer1 2 3 4	260
:NOISe	261
:NOISe:BFACTOR	261
:NOISe:CBWidth	262
:NOISe:CN	262
:NOISe:MUX	262
:REFerence:EXTErnal:FREQUency	263
:REFerence[:SOURce]	263
:RETRigger	264
:RSCALing	264
:SCALing	265
:SCLock:RATE	265
:SEQUence	265
:TRIGger:TYPE	267
:TRIGger:TYPE:CONTInuous[:TYPE]	269
:TRIGger:TYPE:GATE:ACTIve	269
:TRIGger:TYPE:SADVance[:TYPE]	270
:TRIGger[:SOURce]	271
:TRIGger[:SOURce]:EXTErnal[:SOURce]	272
:TRIGger[SOURce]:EXTErnal:DELAy	272
:TRIGger[:SOURce]:EXTErnal:DELAy:STATE	273
:TRIGger[:SOURce]:EXTErnal:SLOPe	273
:VCO:CLOCK	274
:WAVEform	274
:Waveform:NHEAders	275
[:STATE]	275
DmOdulation Subsystem–Option 601 or 602 ([:SOURce]:RADIo:DMODulation:ARB)	276
:IQ:EXTErnal:FILTer	276
:IQ:EXTErnal:FILTer:AUTO	276
:FILTer	277
:FILTer:ALPHA	277
:FILTer:BBT	278
:FILTer:CHANnel	278
:HEADer:CLEAr	279
:HEADer:SAVE	279
:IQ:MODulation:ATTen	279
:IQ:MODulation:ATTen:AUTO	279

---

# Contents

:IQ:MODulation:FILTer . . . . .	280
:IQ:MODulation:FILTer:AUTO . . . . .	280
:MDEStination:ALCHold . . . . .	281
:MDEStination:PULSe . . . . .	282
:MODulation:FSK[:DEVIation] . . . . .	283
:MODulation[:TYPE] . . . . .	284
:MPOLarity:MARKer1 2 3 4 . . . . .	284
:REFerence:EXTErnal:FREQUency . . . . .	285
:REFerence[:SOURce] . . . . .	285
:RETRigger . . . . .	286
:SCLock:RATE . . . . .	286
:SETup . . . . .	287
:SETup:MCARrier . . . . .	287
:SETup:MCARrier:PHASe . . . . .	288
:SETup:MCARrier:STORe . . . . .	288
:SETup:MCARrier:TABLE . . . . .	289
:SETup:MCARrier:TABLE:NCARriers . . . . .	290
:SETup:STORe . . . . .	290
:SRATe . . . . .	290
:TRIGger:TYPE . . . . .	291
:TRIGger:TYPE:CONTInuous[:TYPE] . . . . .	292
:TRIGger:TYPE:GATE:ACTive . . . . .	293
:TRIGger[:SOURce] . . . . .	294
:TRIGger[:SOURce]:EXTErnal[:SOURce] . . . . .	295
:TRIGger[SOURce]:EXTErnal:DELay . . . . .	295
:TRIGger[:SOURce]:EXTErnal:DELay:STATe . . . . .	296
:TRIGger[:SOURce]:EXTErnal:SLOPe . . . . .	296
[:STATe] . . . . .	297
Multitone Subsystem–Option 601 or 602 ([:SOURce]:RADio:MTONE:ARB) . . . . .	298
Creating a Multitone Waveform . . . . .	298
:HEADer:CLEAr . . . . .	298
:HEADer:SAVE . . . . .	298
:IQ:EXTErnal:FILTer . . . . .	298
:IQ:EXTErnal:FILTer:AUTO . . . . .	299
:IQ:MODulation:ATTen . . . . .	299
:IQ:MODulation:ATTen:AUTO . . . . .	300
:IQ:MODulation:FILTer . . . . .	300
:IQ:MODulation:FILTer:AUTO . . . . .	301
:MDEStination:ALCHold . . . . .	301
:MDEStination:PULSe . . . . .	302
:MPOLarity:MARKer1 2 3 4 . . . . .	303

:REfERENCE:EXtERnal:FREQUency . . . . .	304
:REfERENCE[:SOURce]. . . . .	304
:SClock:RATE . . . . .	305
:SETup . . . . .	305
:SETup:STORe . . . . .	306
:SETup:TABLE . . . . .	306
:SETup:TABLE:FSPacing . . . . .	307
:SETup:TABLE:NTONes . . . . .	308
:SETup:TABLE:PHASe:INITialize . . . . .	308
:SETup:TABLE:PHASe:INITialize:SEED . . . . .	309
:ROW . . . . .	309
[:STATe] . . . . .	310
Real Time GPS Subsystem–Option 409	
([:SOURce]:RADio[1] 2 3 4:GPS) . . . . .	311
:DATA . . . . .	311
:DMODE . . . . .	311
:DSHift . . . . .	312
:FILTer . . . . .	312
:FILTer:ALPHa . . . . .	313
:FILTer:BBT . . . . .	314
:FILTer:CHANnel . . . . .	314
:IQPHase . . . . .	315
:PCODE . . . . .	315
:RCODE . . . . .	315
:REFClk . . . . .	316
:REFFreq . . . . .	316
:SATid . . . . .	316
[:STATe] . . . . .	317
Real Time MSGPS Subsystem–Option 409	
([:SOURce]:RADio[1] 2 3 4:MSGPs) . . . . .	318
:IQPHase . . . . .	318
:PLAYmode . . . . .	318
:REFClk . . . . .	319
:REFFreq . . . . .	319
:REStart . . . . .	319
:SCENario . . . . .	320
:SCENario:SATellites . . . . .	320
:SCENario:STATus . . . . .	320
[:STATe] . . . . .	320
Two Tone Subsystem ([:SOURce]:RADio:TTONE:ARB) . . . . .	321
:ALIGNment . . . . .	321

---

## Contents

:APPLY . . . . .	321
:FSPACING . . . . .	321
. . . . .	321
:HEADER:SAVE . . . . .	322
:IQ:EXTERNAL:FILTER . . . . .	322
. . . . .	322
:IQ:MODULATION:ATTEN . . . . .	323
:IQ:MODULATION:ATTEN:AUTO . . . . .	323
:IQ:MODULATION:FILTER . . . . .	324
:IQ:MODULATION:FILTER:AUTO . . . . .	324
:MDESTINATION:ALCHOLD . . . . .	325
:MDESTINATION:PULSE . . . . .	326
:MPOLARITY:MARKER1 2 3 4 . . . . .	327
:REFERENCE:EXTERNAL:FREQUENCY . . . . .	328
:REFERENCE[:SOURCE] . . . . .	328
. . . . .	328
[:STATE]. . . . .	329
Wideband Digital Modulation Subsystem ([:SOURCE]:WDM) . . . . .	330
:IQADJUSTMENT:IOFFSET . . . . .	330
:IQADJUSTMENT:QOFFSET . . . . .	330
:IQADJUSTMENT:QSKEW . . . . .	331
:IQADJUSTMENT[:STATE] . . . . .	331
:STATE . . . . .	331

## 6 Digital Signal Interface Module Commands

Digital Subsystem ([:SOURCE]) . . . . .	333
:DIGITAL:CLOCK:CPS . . . . .	333
:DIGITAL:CLOCK:PHASE . . . . .	334
:DIGITAL:CLOCK:POLARITY . . . . .	334
:DIGITAL:CLOCK:RATE . . . . .	335
:DIGITAL:CLOCK:REFERENCE:FREQUENCY . . . . .	335
:DIGITAL:CLOCK:SOURCE . . . . .	336
:DIGITAL:CLOCK:SKEW . . . . .	336
:DIGITAL:DATA:ALIGNMENT . . . . .	336
:DIGITAL:DATA:BORDER . . . . .	337
:DIGITAL:DATA:DIIRECTION . . . . .	337
:DIGITAL:DATA:IGAIN . . . . .	338
:DIGITAL:DATA:INEGATE . . . . .	338
:DIGITAL:DATA:IOFFSET . . . . .	339
:DIGITAL:DATA:IQSWAP . . . . .	339
:DIGITAL:DATA:NFORMAT . . . . .	339

:DIGital:DATA:POLarity:FRAMe . . . . .	340
:DIGital:DATA:POLarity:IQ . . . . .	340
:DIGital:DATA:QGain . . . . .	341
:DIGital:DATA:QNEGate . . . . .	341
:DIGital:DATA:QOFFset . . . . .	342
:DIGital:DATA:ROTation . . . . .	342
:DIGital:DATA:SCALing . . . . .	342
:DIGital:DATA:SIZE . . . . .	343
:DIGital:DATA:STYPe . . . . .	343
:DIGital:DATA:TYPE. . . . .	344
:DIGital:DIAGnostic:LOOPback. . . . .	344
:DIGital:LOGic[:TYPE] . . . . .	345
:DIGital:PCONfig . . . . .	345
:DIGital:PRESet:PTHROUGH . . . . .	346
:DIGital[:STATe] . . . . .	346

## **7 SCPI Command Compatibility**

:SYSTem:IDN . . . . .	347
E8257D/67D Compatible Commands. . . . .	348
:DATA:PRAM? . . . . .	348
:DATA:PRAM:BLOCK . . . . .	348
:DATA:PRAM:LIST. . . . .	348
:DELeTe:WFM1 . . . . .	348
E8241A/44A/51A/54A and the E8247C/57C/67C PSG Compatible SCPI Commands. . . . .	349
8340B/41B and 8757D Compatible Commands . . . . .	350
836xxB/L Compatible SCPI Commands. . . . .	363
8373xB and 8371xB Compatible SCPI Commands. . . . .	379
8375xB Compatible SCPI Commands . . . . .	387
8662A/63A Compatible Commands . . . . .	399

---

# Contents

# Documentation Overview

## Installation Guide

- Safety Information
- Getting Started
- Operation Verification
- Regulatory Information

## User's Guide

- Signal Generator Overview
- Basic Operation
- Basic Digital Operation
- Optimizing Performance
- Analog Modulation
- Custom Arb Waveform Generator
- Custom Real Time I/Q Baseband
- Multitone Waveform Generator
- Two-Tone Waveform Generator
- AWGN Waveform Generator
- Peripheral Devices
- Troubleshooting

## Programming Guide

- Getting Started with Remote Operation
- Using IO Interfaces
- Programming Examples
- Programming the Status Register System
- Creating and Downloading Waveform Files
- Creating and Downloading User-Data Files

## SCPI Reference

- Using this Guide
- System Commands
- Basic Function Commands
- Analog Commands
- Digital Modulation Commands
- Digital Signal Interface Module Commands
- SCPI Command Compatibility

**Service Guide**

- Troubleshooting
- Replaceable Parts
- Assembly Replacement
- Post-Repair Procedures
- Safety and Regulatory Information

**Key Reference**

- Key function description



---

# 1 Using this Guide

In the following sections, this chapter describes how SCPI information is organized and presented in this guide. An overview of the SCPI language is also provided:

- [“How the SCPI Information is Organized” on page 1](#)
- [“SCPI Basics” on page 2](#)

## How the SCPI Information is Organized

### SCPI Listings

The table of contents lists the Standard Commands for Programmable Instruments (SCPI) without the parameters. The SCPI subsystem name will generally have the first part of the command in parenthesis that is repeated in all commands within the subsystem. The title(s) beneath the subsystem name is the remaining command syntax. The following example demonstrates this listing:

Communication Subsystem (:SYSTem:COMMunicate)

:PMETer:CHANnel

:SERial:ECHO

The following examples show the complete commands from the above Table of Contents listing:

:SYSTem:COMMunicate:PMETer:CHANnel

:SYSTem:COMMunicate:SERial:ECHO

### Subsystem Groupings by Chapter

A subsystem is a group of commands used to configure and operate a certain function or feature. Like individual commands, subsystems that share a similar scope or role can also be categorized and grouped together. This guide uses chapters to divide subsystems into the following groups:

- System Commands
- Basic Function Commands
- Analog Modulation Commands
- Digital Modulation Commands

### Front Panel Operation Cross Reference

The last section in this book provides an index of hardkeys, softkeys, and data fields used in front panel operation, cross-referenced to their corresponding SCPI command. Key and data field names are sorted in two ways:

- individual softkey, hardkey, or data field name
- SCPI subsystem name with associated key and data field names nested underneath

## Supported Models and Options per Command

Within each command section, the Supported heading describes the signal generator configurations supported by the SCPI command. “All” means that all models and options are supported. When “All with Option xxx” is shown next to this heading, only the stated option(s) is supported.

## SCPI Basics

This section describes the general use of the SCPI language for the PSG. It is not intended to teach you everything about the SCPI language; the SCPI Consortium or IEEE can provide that level of detailed information. For a list of the specific commands available for the signal generator, refer to the table of contents.

For additional information, refer to the following publications:

- IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation. New York, NY, 1998.
- IEEE Standard 488.2-1992, IEEE Standard Codes, Formats, Protocols and Command Commands for Use with ANSI/IEEE Standard 488.1-1987. New York, NY, 1998.

## Common Terms

The following terms are used throughout the remainder of this section:

Command	A command is an instruction in SCPI consisting of mnemonics (keywords), parameters (arguments), and punctuation. You combine commands to form messages that control instruments.
Controller	A controller is any device used to control the signal generator, for example a computer or another instrument.
Event Command	Some commands are events and cannot be queried. An event has no corresponding setting; it initiates an action at a particular time.
Program Message	A program message is a combination of one or more properly formatted commands. Program messages are sent by the controller to the signal generator.
Query	A query is a special type of command used to instruct the signal generator to make response data available to the controller. A query ends with a question mark. Generally you can query any command value that you set.
Response Message	A response message is a collection of data in specific SCPI formats sent from the signal generator to the controller. Response messages tell the controller about the internal state of the signal generator.

## Command Syntax

A typical command is made up of keywords prefixed with colons (:). The keywords are followed by parameters. The following is an example syntax statement:

```
[ :SOURce ] :POWer [ :LEVel ] MAXimum|MINimum
```

In the example above, the [ :LEVel ] portion of the command immediately follows the :POWer portion with no separating space. The portion following the [ :LEVel ], MINimum|MAXimum, are the parameters (argument for the command statement). There is a separating space (white space) between the command and its parameter.

Additional conventions in syntax statements are shown in [Table 1-1](#) and [Table 1-2](#).

**Table 1-1 Special Characters in Command Syntax**

Characters	Meaning	Example
	A vertical stroke between keywords or parameters indicates alternative choices. For parameters, the effect of the command varies depending on the choice.	<code>[ :SOURCE]:AM: MOD DEEP NORMAl</code>  DEEP or NORMAl are the choices.
[ ]	Square brackets indicate that the enclosed keywords or parameters are optional when composing the command. These implied keywords or parameters will be executed even if they are omitted.	<code>[ :SOURCE]:FREQuency[:CW]?</code>  SOURCE and CW are optional items.
< >	Angle brackets around a word (or words) indicate they are not to be used literally in the command. They represent the needed item.	<code>[ :SOURCE]:FREQuency: START &lt;val&gt;&lt;unit&gt;</code>  In this command, the words <val> and <unit> should be replaced by the actual frequency and unit.  <code>:FREQuency:START 2.5GHZ</code>
{ }	Braces indicate that parameters can optionally be used in the command once, several times, or not at all.	<code>[ :SOURCE]:LIST: POWER &lt;val&gt;{ ,&lt;val&gt;}</code>  a single power listing: <code>LIST:POWER 5</code> a series of power listings: <code>LIST:POWER 5,10,15,20</code>

**Table 1-2 Command Syntax**

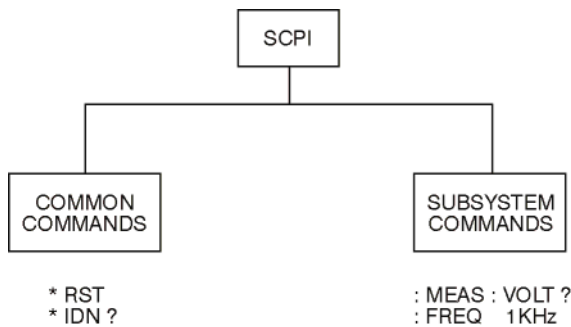
Characters, Keywords, and Syntax	Example
Upper-case lettering indicates the minimum set of characters required to execute the command.	<code>[ :SOURCE]:FREQuency[:CW]?,</code>  FREQ is the minimum requirement.
Lower-case lettering indicates the portion of the command that is optional; it can either be included with the upper-case portion of the command or omitted. This is the flexible format principle called forgiving listening. Refer to <a href="#">"Command Parameters and Responses"</a> on page 5 for more information.	<code>:FREQuency</code>  Either :FREQ, :FREQuency, or :FREQUENCY is correct.
When a colon is placed between two command mnemonics, it moves the current path down one level in the command tree. Refer to <a href="#">"Command Tree"</a> on page 4 more information on command paths.	<code>:TRIGger:OUTPut:POLarity?</code>  TRIGger is the root level keyword for this command.
If a command requires more than one parameter, you must separate adjacent parameters using a comma. Parameters are not part of the command path, so commas do not affect the path level.	<code>[ :SOURCE]:LIST: DWELL &lt;val&gt;{ ,&lt;val&gt;}</code>
A semicolon separates two commands in the same program message without changing the current path.	<code>:FREQ 2.5GHZ;:POW 10DBM</code>
White space characters, such as <tab> and <space>, are generally ignored as long as they do not occur within or between keywords.  However, you must use white space to separate the command from the parameter, but this does not affect the current path.	<code>:FREQ uency</code> or <code>:POWER :LEVel</code> are not allowed.  A <space> between :LEVel and 6.2 is mandatory.  <code>:POWER:LEVel 6.2</code>

## Command Types

Commands can be separated into two groups: common commands and subsystem commands. [Figure 1-1](#), shows the separation of the two command groups. Common commands are used to manage macros, status registers, synchronization, and data storage and are defined by IEEE 488.2. They are easy to recognize because they all begin with an asterisk. For example \*IDN?, \*OPC, and \*RST are common commands. Common commands are not part of any subsystem and the signal generator interprets them in the same way, regardless of the current path setting.

Subsystem commands are distinguished by the colon (:). The colon is used at the beginning of a command statement and between keywords, as in :FREQUency[:CW?]. Each command subsystem is a set of commands that roughly correspond to a functional block inside the signal generator. For example, the power subsystem (:POWer) contains commands for power generation, while the Status subsystem (:STATus) contains commands for controlling status registers.

Figure 1-1 Command Types

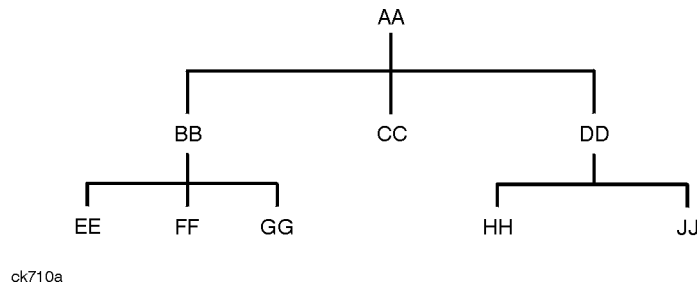


ck709a

## Command Tree

Most programming tasks involve subsystem commands. SCPI uses a structure for subsystem commands similar to the file systems on most computers. In SCPI, this command structure is called a command tree and is shown in [Figure 1-2](#).

Figure 1-2 Simplified Command Tree



The command closest to the top is the root command, or simply “the root.” Notice that you must follow a particular path to reach lower level commands. In the following example, `:POWer` represents AA, `:ALC` represents BB, `:SOURce` represents GG. The complete command path is `:POWer:ALC:SOURce?` (`:AA:BB:GG`).

### Paths Through the Command Tree

To access commands from different paths in the command tree, you must understand how the signal generator interprets commands. The parser, a part of the signal generator firmware, decodes each message sent to the signal generator. The parser breaks up the message into component commands using a set of rules to determine the command tree path used. The parser keeps track of the current path (the level in the command tree) and where it expects to find the next command statement. This is important because the same keyword may appear in different paths. The particular path is determined by the keyword(s) in the command statement.

A message terminator, such as a `<new line>` character, sets the current path to the root. Many programming languages have output statements that automatically send message terminators.

---

**NOTE** The current path is set to the root after the line–power is cycled or when `*RST` is sent.

---

### Command Parameters and Responses

SCPI defines different data formats for use in program and response messages. It does this to accommodate the principle of forgiving listening and precise talking. For more information on program data types refer to IEEE 488.2. Forgiving listening means the command and parameter formats are flexible.

For example, with the `:FREQuency:REFerence:STATe ON|OFF|1|0` command, the signal generator accepts `:FREQuency:REFerence:STATe ON`, `:FREQuency:REFerence:STATe 1`, `:FREQ:REF:STAT ON`, `:FREQ:REF:STAT 1` to turn on the frequency reference mode.

Each parameter type has one or more corresponding response data types. A setting that you program using a numeric parameter returns either real or integer response data when queried. Response data (data returned to the controller) is more concise and restricted and is called precise talking.

Precise talking means that the response format for a particular query is always the same.

For example, if you query the power state (`:POWer:ALC:STATe?`) when it is on, the response is always 1, regardless of whether you previously sent `:POWer:ALC:STATe 1` or `:POWer:ALC:STATe ON`.

**Table 1-3 Parameter and Response Types**

Parameter Types	Response Data Types
Numeric	Real, Integer
Extended Numeric	Real, Integer
Discrete	Discrete
Boolean	Numeric Boolean
String	String

### Numeric Parameters

Numeric parameters are used in both common and subsystem commands. They accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation.

If a signal generator setting is programmed with a numeric parameter which can only assume a finite value, it automatically rounds any entered parameter which is greater or less than the finite value. For example, if a signal generator has a programmable output impedance of 50 or 75 ohms, and you specified 76.1 for the output impedance, the value is rounded to 75. The following are examples of numeric parameters:

100	no decimal point required
100.	fractional digits optional
-1.23	leading signs allowed
4.56E<space>3	space allowed after the E in exponential
-7.89E-001	use either E or e in exponential
+256	leading + allowed
.5	digits left of decimal point optional

### Extended Numeric Parameters

Most subsystems use extended numeric parameters to specify physical quantities. Extended numeric parameters accept all numeric parameter values and other special values as well.

The following are examples of extended numeric parameters:

Extended Numeric Parameters		Special Parameters	
100	any simple numeric value	DEFault	resets parameter to its default value
1.2GHZ	GHZ can be used for exponential (E009)	UP	increments the parameter
200MHZ	MHZ can be used for exponential (E006)	DOWN	decrements the parameter

Extended Numeric Parameters		Special Parameters	
-100mV	negative 100 millivolts	MINimum	sets parameter to smallest possible value
10DEG	10 degrees	MAXimum	sets parameter to largest possible value

## Discrete Parameters

Discrete parameters use mnemonics to represent each valid setting. They have a long and a short form, just like command mnemonics. You can mix upper and lower case letters for discrete parameters.

The following examples of discrete parameters are used with the command `:TRIGger[:SEquence]:SOURce BUS|IMMediate|EXTernal`.

BUS	GPIB, LAN, or RS-232 triggering
IMMediate	immediate trigger (free run)
EXTernal	external triggering

Although discrete parameters look like command keywords, do not confuse the two. In particular, be sure to use colons and spaces correctly. Use a colon to separate command mnemonics from each other and a space to separate parameters from command mnemonics.

The following are examples of discrete parameters in commands:

```
TRIGger:SOURce BUS
TRIGger:SOURce IMMediate
TRIGger:SOURce EXTernal
```

## Boolean Parameters

Boolean parameters represent a single binary condition that is either true or false. The two-state boolean parameter has four arguments. The following list shows the arguments for the two-state boolean parameter:

ON	boolean true, upper/lower case allowed
OFF	boolean false, upper/lower case allowed
1	boolean true
0	boolean false

### String Parameters

String parameters allow ASCII strings to be sent as parameters. Single or double quotes are used as delimiters.

The following are examples of string parameters:

```
'This is valid'           "This is also valid"           'SO IS THIS'
```

### Real Response Data

Real response data represent decimal numbers in either fixed decimal or scientific notation. Most high-level programming languages that support signal generator input/output (I/O) handle either decimal or scientific notation transparently.

The following are examples of real response data:

```
+4.000000E+010, -9.990000E+002  
-9.990000E+002  
+4.000000000000000E+010  
+1  
0
```

### Integer Response Data

Integer response data are decimal representations of integer values including optional signs. Most status register related queries return integer response data. The following are examples of integer response data:

```
0      signs are optional           -100   leading - allowed  
+100   leading + allowed           256    never any decimal point
```

### Discrete Response Data

Discrete response data are similar to discrete parameters. The main difference is that discrete response data only returns the short form of a particular mnemonic, in all upper case letters. The following are examples of discrete response data:

```
IMM      EXT      INT      NEG
```

### Numeric Boolean Response Data

Boolean response data returns a binary numeric value of one or zero.



## String Response Data

String response data are similar to string parameters. The main difference is that string response data returns double quotes, rather than single quotes. Embedded double quotes may be present in string response data. Embedded quotes appear as two adjacent double quotes with no characters between them. The following are examples of string response data:

```
"This is a string"
"one double quote inside brackets: ["]"
"Hello!"
```

## Program Messages

The following commands will be used to demonstrate the creation of program messages:

```
[ :SOURce]:FREQuency:START           [ :SOURce]:FREQuency:STOP
[ :SOURce]:FREQuency[:CW]           [ :SOURce]:POWer[:LEVel]:OFFSet
```

### Example 1

```
:FREQuency:START 500MHZ;STOP 1000MHZ
```

This program message is correct and will not cause errors; START and STOP are at the same path level. It is equivalent to sending the following message:

```
FREQuency:START 500MHZ;FREQuency:STOP 1000MHZ
```

### Example 2

```
:POWer 10DBM;:OFFSet 5DB
```

This program message will result in an error. The message makes use of the default POWER[:LEVel] node (root command). When using a default node, there is no change to the current path position. Since there is no command OFFSet at the root level, an error results.

The following example shows the correct syntax for this program message:

```
:POWer 10DBM;:POWer:OFFSet 5DB
```

### Example 3

```
:POWer:OFFSet 5DB;POWer 10DBM
```

This program message results in a command error. The path is dropped one level at each colon. The first half of the message drops the command path to the lower level command OFFSet; POWER does not exist at this level.

The POWER 10DBM command is missing the leading colon and when sent, it causes confusion because the signal generator cannot find POWER at the POWER:OFFSet level. By adding the leading colon, the current path is reset to the root. The following shows the correct program message:

```
:POWer:OFFSet 5DB;:POWer 10DBM
```

#### Example 4

```
FREQ 500MHZ;POW 4DBM
```

In this example, the keyword short form is used. The program message is correct because it utilizes the default nodes of :FREQ[:CW] and :POW[:LEVEl]. Since default nodes do not affect the current path, it is not necessary to use a leading colon before FREQ or POW.

#### File Name Variables

File name variables, such as "<file name>", represent three formats, "<file name>", "<file name@file type>", and "</user/file type/file name>". The following shows the file name syntax for the three formats, but uses "FLATCAL" as the file name in place of the variable "<file name>":

```
Format 1      "FLATCAL"
Format 2      "FLATCAL@USERFLAT"
Format 3      "/USER/USERFLAT/FLATCAL"
```

Format 2 uses the file type extension (@USERFLAT) as part of the file name syntax. Format 3 uses the directory path which includes the file name and file type. Use Formats 2 and 3 when the command does not specify the file type. This generally occurs in the Memory (:MEMoRY) or Mass Memory (:MMEMoRY) subsystems. The following examples demonstrate a command where Format 1 applies:

```
Command Syntax with the file name variable      :MEMoRY:SToRE:LIST "<file name>"
Command Syntax with the file name              :MEMoRY:SToRE:LIST "SWEEP_1"
```

This command has :LIST in the command syntax. This denotes that "SWEEP\_1" will be saved in the :List file type location as a list type file.

The following examples demonstrate a command where Format 2 applies:

```
Command Syntax with the file name variable
:MMEMoRY:CoPY "<filename>","<filename>"

Command Syntax with the file name
:MMEMoRY:CoPY "FLATCAL@USERFLAT","FLAT_2CAL@USERFLAT"
```

This command cannot distinguish which file type "FLATCAL" belongs to without the file type extension (@USERFLAT). If this command were executed without the extension, the command would assume the file type was Binary.

The following examples demonstrate a command where format 3 applies:

*Command Syntax with the file name variable*

```
:MMEMory:DATA "/USER/BBG1/WAVEFORM/<file name>",#ABC
```

*Command Syntax with the file name*

```
:MMEMory:DATA "/USER/BBG1/WAVEFORM/FLATCAL",#ABC
```

This command gives the directory path name where the file "FLATCAL" is stored.

- |   |  |
|---|--|
| A | the number of decimal digits to follow in B.               |
| B | a decimal number specifying the number of data bytes in C. |
| C | the binary waveform data.                                  |

Refer to [Table 2-1 on page 59](#) for a listing of the file systems and types. The entries under file type are used in the directory path.

## ARB Waveform File Directories

ARB waveform files can be saved to the following directories:

- WFM1: volatile ARB waveform storage. Files located here can be played by the signal generator's arb player, but are volatile and will be lost on a power cycle. The directory can also be specified as /USER/BBG1/WAVEFORM.
- NVWFM: non-volatile ARB waveform storage. Files must be moved to the WFM1: directory before they can be played by the signal generator's Dual ARB player. The directory can also be specified as /USER/WAVEFORM.
- SEQ: sequence files are stored here and are non-volatile. The directory can also be specified as /USER/SEQ.

## MSUS (Mass Storage Unit Specifier) Variable

The variable "<msus>" enables a command to be file type specific when working with user files. Some commands use it as the only command parameter, while others can use it in conjunction with a file name when a command is not file type specific. When used with a file name, it is similar to Format 2 in the [File Name Variables](#) section on [page 10](#). The difference is the file type specifier (msus) occupies its own variable and is not part of the file name syntax.

The following examples illustrate the usage of the variable "<msus>" when it is the only command parameter:

*Command Syntax with the msus variable*

```
:MMEMory:CATalog? "<msus>"
```

*Command Syntax with the file system*

```
:MMEMory:CATalog? "LIST:"
```

The variable "<msus>" is replaced with "LIST:". When the command is executed, the output displays only the files from the List file system.

The following examples illustrate the usage of the variable "<file name>" with the variable "<msus>":

*Command Syntax with the file name and msus variables*

```
:MMEMory:DElete[:NAME] "<file name>","<msus>"
```

*Command Syntax with the file name and file system*

```
:MMEMory:DElete:NAME "LIST_1","LIST:"
```

The command from the above example cannot discern which file system LIST\_1 belongs to without a file system specifier and will not work without it. When the command is properly executed, LIST\_1 is deleted from the List file system.

The following example shows the same command, but using Format 2 from the [File Name Variables](#) section on [page 10](#):

```
:MMEMory:DElete:NAME "LIST_1@LIST"
```

When a file name is a parameter for a command that is not file system specific, either format (<file name>,"<msus>" or "<file name@file system>") will work.

Refer to [Table 1-1 on page 3](#) for a listing of special syntax characters.

## Quote Usage with SCPI Commands

As a general rule, programming languages require that SCPI commands be enclosed in double quotes as shown in the following example:

```
":FM:EXtErnal:IMPedance 600"
```

However, when a string is the parameter for a SCPI command, additional quotes or other delimiters may be required to identify the string. Your programming language may use two sets of double quotes, one set of single quotes, or back slashes with quotes to signify the string parameter. The following examples illustrate these different formats:

```
"MEMory:LOAD:LIST "myfile"" used in BASIC programming languages
```

```
"MEMory:LOAD:LIST \"myfile\" used in C, C++, Java, and PERL
```

```
"MEMory:LOAD:LIST 'myfile' accepted by most programming languages
```

Consult your programming language reference manual to determine the correct format.

## Binary, Decimal, Hexadecimal, and Octal Formats

Command values may be entered using a binary, decimal, hexadecimal, or octal format. When the binary, hexadecimal, or octal format is used, their values must be preceded with the proper identifier. The decimal format (default format) requires no identifier and the signal generator assumes this format when a numeric value is entered without one. The following list shows the identifiers for the formats that require them:

- #B identifies the number as a binary numeric value (base-2).
- #H identifies the number as a hexadecimal alphanumeric value (base-16).
- #Q identifies the number as a octal alphanumeric value (base-8).

The following are examples of SCPI command values and identifiers for the decimal value 45:

```
#B101101      binary equivalent
#H2D          hexadecimal equivalent
#Q55          octal equivalent
```

The following example sets the RF output power to 10 dBm (or the equivalent value for the currently selected power unit, such as DBUV or DBUVEMF) using the hexadecimal value 000A:

```
:POW #H000A
```

A unit of measure, such as DBM or mV, will not work with the values when using a format other than decimal.

The following example sets the bluetooth board address to FFBF7 (hexadecimal):

```
:RADio:BLUETooth:ARB:BDADdr #HFFBF7
```



---

## 2 System Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to peripheral signal generator operations common to all PSG models:

- “Calibration Subsystem (:CALibration)” on page 16
- “Communication Subsystem (:SYSTem:COMMunicate)” on page 21
- “Diagnostic Subsystem (:DIAGnostic[:CPU]:INFORmation)” on page 28
- “Display Subsystem (:DISPlay)” on page 30
- “IEEE 488.2 Common Commands” on page 34
- “Low-Band Filter Subsystem” on page 39
- “Memory Subsystem (:MEMory)” on page 39
- “Mass Memory Subsystem (:MMEMory)” on page 59
- “Output Subsystem (:OUTPut)” on page 65
- “Route Subsystem (:ROUte:HARDware:DGENerator)” on page 69
- “Status Subsystem (:STATus)” on page 76
- “System Subsystem (:SYSTem)” on page 90
- “Trigger Subsystem” on page 105
- “Unit Subsystem (:UNIT)” on page 108

## Calibration Subsystem (:CALibration)

### :DCFM

**Supported** All with Option UNT

:CALibration:DCFM

This command initiates a DCFM or DCΦM calibration depending on the currently active modulation. This calibration eliminates any dc or modulation offset of the carrier signal.

Use this calibration for externally applied signals. While the calibration can also be performed for internally generated signals, dc offset is not a normal characteristic for them.

---

**NOTE** If the calibration is performed with a dc signal applied, any deviation provided by the dc signal will be removed and the new zero reference point will be at the applied dc level. The calibration will have to be performed again when the dc signal is removed in order to reset the carrier signal to the correct zero reference.

---

**Key Entry** DCFM/DCΦM Cal

### :IQ

**Supported** E8267D

:CALibration:IQ

This command initiates an I/Q calibration for a range of frequencies and is equivalent to selecting **User** from the front panel **Calibration Type DC User Full** softkey in the I/Q Calibration menu. For setting range frequencies, refer to :IQ:START, and :IQ:STOP commands.

**Key Entry** Execute Cal Calibration Type DC User Full

### :IQ:DC

**Supported** E8267D

:CALibration:IQ:DC

This command starts and performs a one to two second adjustment that is not traceable to a standard. However, it will minimize errors associated with signal generator internal voltage offsets. This adjustment minimizes errors for the current signal generator setting and at a single frequency. The DC adjustment is volatile and must be repeated with each signal generator setting change. This command can be sent while the RF On/Off is set to Off and the adjustment will still be valid when the RF is enabled.

The I/Q DC adjustment is dependent upon a number of instrument settings. If any of the instrument settings change, the adjustment will become invalid. The dependent instrument settings are:

- RF frequency
- I/Q attenuation level
- Baseband generator settings



- I/Q polarity settings
- Baseband filter settings
- Path settings (Internal I/Q Mux Path 1 or Path 2)
- I/Q calibration (the I/Q DC calibration will be invalidated if any other I/Q calibration is execute)
- Temperature ( $\pm 5$  degrees)

The following instrument states will not invalidate the I/Q DC calibration:

- Power level changes
- I/Q Impairments

**Key Entry**            **Execute Cal**        **Calibration Type DC User Full**

### **:IQ:DEfault**

**Supported**            E8267D

:CALibration:IQ:DEfault

This command will restore the original factory calibration data for the internal I/Q modulator.

**Key Entry**            **Revert to Default Cal Settings**

### **:IQ:FULL**

**Supported**            E8267D

:CALibration:IQ:FULL

This command sets and performs a full frequency range (regardless of the start and stop frequency settings) I/Q calibration and stores the results in the signal generator's memory.

Start and stop frequencies default to the full frequency range of the signal generator.

**Range**                Depends on the signal generator's frequency option.  
See also: "[:FREQUENCY:CENTer](#)" on page 113.

**Key Entry**            **Execute Cal** (**Calibration Type DC User Full** set to Full)

## :IQ:STARt

**Supported** E8267D

```
:CALibration:IQ:STARt <val><units>  
:CALibration:IQ:STARt?
```

This command sets the start frequency and automatically sets the calibration type to User for an I/Q calibration.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:CAL:IQ:STAR 1GHZ
```

The preceding example sets the signal generator's start frequency for an IQ calibration to 1 GHz.

**Range** Depends on the signal generator's frequency option.  
See also: [“.FREQUENCY:CENTer” on page 113.](#)

**Key Entry** **Start Frequency**

## :IQ:STOP

**Supported** E8267D

```
:CALibration:IQ:STOP <val><units>  
:CALibration:IQ:STOP?
```

This command sets the stop frequency and automatically sets the calibration type to User for an I/Q calibration. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:CAL:IQ:STOP 2GHZ
```

The preceding example sets the signal generator's stop frequency for an IQ calibration to 2 GHz.

**Range** Depends on the signal generator's frequency option.  
See also: [“.FREQUENCY:CENTer” on page 113.](#)

**Key Entry** **Stop Frequency**

## :WBIQ

**Supported** E8267D with Option 015

```
:CALibration:WBIQ
```

This command initiates a wideband I/Q calibration for a range of frequencies and is equivalent to selecting **User** from the front panel **Calibration Type DC User Full** softkey. For setting range frequencies, refer to [:WBIQ:STARt](#), and [:WBIQ:STOP](#) commands.

**Key Entry** **Execute Cal**

## **:WBIQ:DC**

**Supported** E8267D with Option 015

:CALibration:WBIQ:DC

This command performs a one to two second adjustment that is not traceable to a standard. However, it will minimize errors associated with offset voltages. This adjustment minimizes errors for the current signal generator setting and at a single frequency. The DC adjustment is volatile and must be repeated with each signal generator setting change. This command can be sent while the RF On/Off is set to Off and the adjustment will be valid when RF is enabled.

The wideband I/Q DC adjustment is dependent upon a number of instrument settings. If any of the PSG settings change, the adjustment will become invalid. The dependent instrument settings are:

- RF frequency
- I/Q attenuation level
- Baseband generator settings
- I/Q polarity settings
- Baseband filter settings
- Path settings (Internal I/Q Mux Path 1 or Path 2)
- I/Q calibration (the I/Q DC calibration will be invalidated if any other I/Q calibration is executed)
- Temperature ( $\pm 5$  degrees)

The following instrument states will not invalidate the I/Q DC calibration:

- Power level changes
- I/Q Impairments

<b>Key Entry</b>	<b>Execute Cal</b>	<b>Calibration Type DC User Full</b>
------------------	--------------------	--------------------------------------

## **:WBIQ:DEFault**

**Supported** E8267D with Option 015

:CALibration:WBIQ:DEFault

This command will restore the original factory calibration data for the internal I/Q modulator.

<b>Key Entry</b>	<b>Revert to Default Cal Settings</b>
------------------	---------------------------------------

## **:WBIQ:FULL**

**Supported** E8267D with Option 015

:CALibration:WBIQ:FULL

This command sets and performs a full-frequency range (regardless of the start and stop frequency settings) wideband I/Q calibration and stores the results in the signal generator's firmware.

Start and stop frequencies will default to the full frequency range of the signal generator.

**Range** Depends on the signal generator's frequency option.  
See also: "[:FREQUENCY:CENTer](#)" on page 113.

**Key Entry** **Execute Cal**      **Calibration Type DC User Full**

## **:WBIQ:STARt**

**Supported** E8267D with Option 015

:CALibration:WBIQ:STARt <val><units>

:CALibration:WBIQ:STARt?

This command sets the start frequency and automatically sets the calibration type to User for a wideband I/Q calibration. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:CAL:WBIQ:STAR 1GHZ
```

The preceding example sets the signal generator's start frequency to 1 GHz for a wideband IQ calibration.

**Range** Depends on the signal generator's frequency option.  
See also: "[:FREQUENCY:CENTer](#)" on page 113.

**Key Entry** **Start Frequency**

## **:WBIQ:STOP**

**Supported** E8267D with Option 015

:CALibration:WBIQ:STOP <val><units>

:CALibration:WBIQ:STOP?

This command sets the stop frequency and automatically sets the calibration type to User for a wideband I/Q calibration.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:CAL:WBIQ:STOP 2GHZ
```

The preceding example sets the signal generator's stop frequency to 2 GHz for a wideband IQ calibration.

**Range** Depends on the signal generator's frequency option.  
See also: [“.FREQuency:CENTer” on page 113.](#)

**Key Entry** **Stop Frequency**

## Communication Subsystem (:SYSTem:COMMunicate)

### :GPIB:ADDRess

**Supported** All Models

```
:SYSTem:COMMunicate:GPIB:ADDRess <number>
```

```
:SYSTem:COMMunicate:GPIB:ADDRess?
```

This command sets the signal generator's general purpose instrument bus (GPIB) address.

The variable <number> is a numeric value between 0 and 30. The signal generator typically uses 19 as the instrument address. The address must be different from other GPIB devices in your system.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:SYST:COMM:GPIB:ADDR 19
```

The preceding example sets the signal generator's GPIB address to 19.

**Range** 0–30

**Key Entry** **GPIB Address**

### :GTLocal

**Supported** All Models

```
:SYSTem:COMMunicate:GTLocal
```

This command sets the signal generator to local mode, enabling front panel operation.

**Range** N/A

**Key Entry** **Local**

## **:LAN:CONFig**

**Supported**            All Models

```
:SYSTem:COMMunicate:LAN:CONFig DHCP|MANual  
:SYSTem:COMMunicate:LAN:CONFig?
```

This command selects the signal generator's internet protocol (IP) address. The dynamic host communication protocol (DHCP) selection allows the network to assign an IP address. The manual selection allows the user to enter an IP address.

### **Example**

```
:SYST:COMM:LAN:CONF DHCP
```

The preceding example sets up the signal generator LAN configuration to use a DHCP IP address.

**Key Entry**            **LAN Config**

## **:LAN:GATEway**

**Supported**            All Models

```
:SYSTem:COMMunicate:LAN:GATEway "<ipstring>"  
:SYSTem:COMMunicate:LAN:GATEway?
```

This command sets the gateway for local area network (LAN) access to the signal generator from outside the current sub-network.

The "<ipstring>" string variable is the LAN gateway address, formatted as xxx.xxx.xxx.xxx. Refer to [Quote Usage with SCPI Commands](#) for information on using quotes for different programming languages.

Using an empty string restricts access to the signal generator to local hosts on the LAN.

### **Example**

```
:SYST:COMM:LAN:GATE "203.149.781.101"
```

The preceding example sets the signal generator's LAN gateway address.

**Key Entry**            **Default Gateway**

## **:LAN:HOSTname**

**Supported**            All Models

```
:SYSTem:COMMunicate:LAN:HOSTname "<string>"  
:SYSTem:COMMunicate:LAN:HOSTname?
```

This command sets the signal generator's local area network (LAN) connection hostname.

The "<string>" variable is the hostname for the signal generator. Refer to [Quote Usage with SCPI Commands](#) for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:SYST:COMM:LAN:HOSTname "siginst3"
```

The preceding example sets “siginst3” as the signal generator’s LAN hostname.

**Key Entry**           **Hostname**

### :LAN:IP

**Supported**           All Models

```
:SYSTem:COMMunicate:LAN:IP "<ipstring>"  
:SYSTem:COMMunicate:LAN:IP?
```

This command sets the signal generator’s local area network (LAN) internet protocol (IP) address for your IP network connection.

The "<ipstring>" variable is the signal generator’s IP address, formatted as xxx.xxx.xxx.xxx. Refer to [Quote Usage with SCPI Commands](#) for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:SYST:COMM:LAN:IP "202.195.207.193"
```

The preceding example sets the signal generator’s LAN IP address.

**Key Entry**           **IP Address**

### :LAN:SUBNet

**Supported**           All Models

```
:SYSTem:COMMunicate:LAN:SUBNet "<ipstring>"  
:SYSTem:COMMunicate:LAN:SUBNet?
```

This command sets the signal generator’s local area network (LAN) subnet mask address for your internet protocol (IP) network connection.

The "<ipstring>" variable is the subnet mask for the IP address, formatted as xxx.xxx.xxx.xxx. Refer to [Quote Usage with SCPI Commands](#) for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:SYST:COMM:LAN:SUBN "203.194.101.111"
```

The preceding example sets the signal generator’s LAN subnet mask.

**Key Entry**           **Subnet Mask**

## **:PMETer:ADDRes**

**Supported**            All Models

```
:SYSTem:COMMunicate:PMETer:ADDRes <val>  
:SYSTem:COMMunicate:PMETer:ADDRes?
```

This command sets the instrument address for a power meter that is controlled by the signal generator. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

The variable <number> is an integer numeric value between 0 and 30. The power meter address must be different from the GPIB address of the signal generator and any other GPIB instrument addresses in your system.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:SYST:COMM:PMET:ADDR 14
```

The preceding example sets the address to 14 for the power meter that is connected to and controlled by the signal generator.

**Range**                0–30

**Key Entry**            **Meter Address**

## **:PMETer:CHANnel**

**Supported**            All Models

```
:SYSTem:COMMunicate:PMETer:CHANnel A|B  
:SYSTem:COMMunicate:PMETer:CHANnel?
```

This command sets the measurement channel on a dual channel power meter that is controlled by the signal generator. A single-channel power meter uses channel A and selecting channel B will have no effect.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

### **Example**

```
:SYST:COMM:PMET:CHAN B
```

The preceding example sets the B measurement channel for the power meter that is connected to and controlled by the signal generator.

**Key Entry**            **Meter Channel A B**



## **:PMETer:IDN**

**Supported**            All Models

```
:SYSTem:COMMunicate:PMETer:IDN E4418B|E4419B|E4416A|E4417A
:SYSTem:COMMunicate:PMETer:IDN?
```

This command sets the model number of the power meter that is controlled by the signal generator. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

### **Example**

```
:SYST:COMM:PMET:IDN E4417A
```

The preceding example sets the model number for the power meter that is connected to and controlled by the signal generator.

**Key Entry**            **Power Meter**

## **:PMETer:TIMEout**

**Supported**            All Models

```
:SYSTem:COMMunicate:PMETer:TIMEout <num>[<time_suffix>]
:SYSTem:COMMunicate:PMETer:TIMEout?
```

This command sets the period of time that the signal generator will wait for a valid reading from the power meter. The variable <num> has a resolution of 0.001.

The variable <num> is the time expressed as a number. The variable <time\_suffix> are the units of time, for example mS (milliseconds) or S (seconds).

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable. If a timeout occurs, the signal generator reports an error message.

### **Example**

```
:SYST:COMM:PMET:TIME .1SEC
```

The preceding example sets the timeout to 100 milliseconds for the power meter that is connected to and controlled by the signal generator.

**Range**                1 mS–100 seconds

**Key Entry**            **Meter Timeout**

## **:SERial:BAUD**

**Supported**            All Models

```
:SYSTem:COMMunicate:SERial:BAUD <number>  
:SYSTem:COMMunicate:SERial:BAUD?
```

This command sets the baud rate for the rear panel RS-232 interface labeled RS-232. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

The variable <number> is an integer value corresponding to baud rates: 300, 2400, 4800, 9600, 19200, 38400, and 57600.

### **Example**

```
:SYST:COMM:SER:BAUD 9600
```

The preceding example sets the baud rate for serial communication to 9600.

**Key Entry**            **RS-232 Baud Rate**

## **:SERial:ECHO**

**Supported**            All Models

```
:SYSTem:COMMunicate:SERial:ECHO ON|OFF  
:SYSTem:COMMunicate:SERial:ECHO?
```

This command enables or disables the RS-232 echo, and is not affected by a power-on, preset, or \*RST command. Characters sent to the signal generator are displayed or echoed to the controller display.

### **Example**

```
:SYST:COMM:SER:ECHO ON
```

The preceding example enables RS-232 echoing.

**Key Entry**            **RS-232 ECHO Off On**

## **:SERial:RESet**

**Supported**            All Models

```
:SYSTem:COMMunicate:SERial:RESet
```

This event command resets the RS-232 buffer and discards unprocessed SCPI input received at the RS-232 port.

**Key Entry**            **Reset RS-232**

## **:SERial:TOUT**

**Supported**            All Models

```
:SYSTem:COMMunicate:SERial:TOUT <val>  
:SYSTem:COMMunicate:SERial:TOUT?
```

This command sets the RS-232 serial port timeout value. If further input is not received within the timeout period specified while a SCPI command is processed, the command aborts and clears the input buffer. The variable <val> is entered in seconds. The setting is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:SYST:COMM:SER:TOUT 2SEC
```

The preceding example sets the RS-232 timeout for 2 seconds.

**Range**                1-25

**Key Entry**            **RS- 232 Timeout**

## Diagnostic Subsystem (:DIAGnostic[:CPU]:INFORMATION)

### :BOARDs

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:BOARDs?

This query returns a list of the boards installed in the signal generator. The information is returned in the following format:

"<board\_name,part\_number,serial\_number,version\_number,status>"

This information format will repeat for each of the signal generator's detected boards.

**Key Entry** **Installed Board Info**

### :CCOunt:ATTenuator

**Supported** E8257D/E8663D with Option 1E1 and E8267D

:DIAGnostic[:CPU]:INFORMATION:CCOunt:ATTenuator?

This query returns the cumulative number of times that the attenuator has switched levels.

**Key Entry** **Diagnostic Info**

### :CCOunt:PON

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:CCOunt:PON?

This query returns the cumulative number of times the signal generator has been powered-on.

**Key Entry** **Diagnostic Info**

### :DISPlay:OTIME

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:DISPlay:OTIME?

This query returns the cumulative number of hours the display has been on.

**Key Entry** **Diagnostic Info**

### :LICENSe:AUXiliary

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:LICENSe:AUXiliary?

This query returns a listing of current external software application license numbers for an auxiliary instrument.

**Key Entry** **Auxiliary Software Options**

## **:OPTions**

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:OPTions?

This query returns a list of options installed in the signal generator.

**Key Entry**           **Options Info**

## **:OPTions:DETail**

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:OPTions:DETail?

This query returns the options installed, option revision, and digital signal processing (DSP) version if applicable.

**Key Entry**           **Options Info**

## **:OTIME**

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:OTIME?

This query returns the cumulative number of hours that the signal generator has been on.

**Key Entry**           **Diagnostic Info**

## **:REVision**

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:REVision?

This query returns the CPU bootstrap read only memory (boot ROM) revision date. In addition, the query returns the revision, creation date, and creation time for the firmware.

**Key Entry**           **Diagnostic Info**

## **:SDATe**

**Supported** All Models

:DIAGnostic[:CPU]:INFORMATION:SDATe?

This query returns the date and time stamp for the signal generator's firmware.

**Key Entry**           **Diagnostic Info**

## Display Subsystem (:DISPlay)

### :ANNotation:AMPLitude:UNIT

Supported            All Models

```
:DISPlay:ANNotation:AMPLitude:UNIT DBM|DBUV|DBUVEMF|V|VEMF|DB  
:DISPlay:ANNotation:AMPLitude:UNIT?
```

This command sets the displayed front panel amplitude units.

If the amplitude reference state is set to on, the query returns units expressed in dB. Setting any other unit will cause a setting conflict error stating that the amplitude reference state must be set to off. Refer to [:REFERENCE:STATE](#) command for more information.

#### Example

```
:DISP:ANN:AMPL:UNIT DB
```

The preceding example sets DB as the amplitude units shown on the signal generator's front panel display.

```
*RST                dBm
```

### :ANNotation:CLOCK:DATE:FORMat

Supported            All Models

```
:DISPlay:ANNotation:CLOCK:DATE:FORMat MDY|DMY  
:DISPlay:ANNotation:CLOCK:DATE:FORMat?
```

This command selects the date format. The choices are month-day-year (MDY) or day-month-year (DMY) format. The date is shown on the signal generator's front panel display.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

#### Example

```
:DISP:ANN:CLOC:DATA:FORM DMY
```

The preceding example sets the date format shown on the signal generator's front panel display to DMY.

### **:ANNotation:CLOCK[:STATe]**

Supported            All Models

```
:DISPlay:ANNotation:CLOCK[:STATe] ON|OFF|1|0  
:DISPlay:ANNotation:CLOCK[:STATe]?
```

This command enables or disables the digital clock shown at the lower right side of the front panel display.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

#### **Example**

```
:DISP:ANN:CLOC OFF
```

The preceding example disables the digital clock on the signal generator's front panel display.

### **:BRIGhtness**

Supported            All Models

```
:DISPlay:BRIGhtness <val>  
:DISPlay:BRIGhtness?
```

This command sets the display brightness (intensity). The brightness can be set to the minimum level (0.02), maximum level (1), or in between by using fractional numeric values (0.03–0.99).

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

#### **Example**

```
:DISP:BRIG .45
```

The preceding example sets display intensity to .45.

**Range**                0.02–1

**Key Entry**            **Brightness**

## :CAPTure

**Supported**            All Models

:DISPlay:CAPTure

This command allows the user to capture the current display and store it in the signal generator's memory.

The display capture is stored as DISPLAY.BMP in the Binary file system. This file is overwritten with each subsequent display capture. The file can be down-loaded in the following manner:

1. Log on to the signal generator using file transfer protocol (FTP).
2. Change to the BIN directory using the FTP cd command.
3. Retrieve the file by using the FTP get command.

## :CONTrast

**Supported**            All Models

:DISPlay:CONTrast <val>

:DISPlay:CONTrast?

This command sets the contrast for the signal generator's display. The variable <val> is expressed as a fractional number between 0 and 1. The contrast can be set to the maximum level (1), minimum level (0), or in between by using fractional numeric values (0.001–0.999).

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:DISP:CONT .45
```

The preceding example sets the display contrast to .45.

**Range**                0–1

**Key Entry**            Display contrast hardkeys are located below the display.

## :INVerse

**Supported**            All Models

:DISPlay:INVerse ON|OFF|1|0

:DISPlay:INVerse?

This command sets the display of the source to inverse video mode. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:DISP:INV OFF
```



The preceding example sets the display video to normal (not inverse).

**Key Entry**            **Inverse Video Off On**

**:REMOte**

**Supported**            All Models

```
:DISPlay:REMOte ON|OFF|1|0
:DISPlay:REMOte?
```

This command enables or disables display updating when the signal generator is remotely controlled.

ON (1)                This choice updates the signal generator display so that you can see the settings change as the commands are executed, however, this will decrease the signal generator's response time.

OFF (0)              This choice turns off display updating which will optimizing the signal generator's response time.

The setting enabled by this command is not affected by signal generator preset or \*RST command. However, cycling the signal generator power will reset it to zero.

**Example**

```
:DISP:REM 0
```

The preceding example turns off display updating.

**Key Entry**            **Update in Remote Off On**

**[:WINDow][:STATe]**

**Supported**            All Models

```
:DISPlay[:WINDow][:STATe] ON|OFF|1|0
:DISPlay[:WINDow][:STATe]?
```

This command is used to either blank out (OFF or 0) the display screen or turn it on (ON or 1).

A signal generator preset, \*RST command, or cycling the power will turn the display on.

**Example**

```
:DISP OFF
```

The preceding example blanks out the signal generator's display.

## IEEE 488.2 Common Commands

### \*CLS

**Supported**            All Models

\*CLS

The Clear Status (CLS) command clears the Status Byte register, the Data Questionable Event register, the Standard Event Status register, and the Standard Operation Status register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### \*ESE

**Supported**            All Models

\*ESE <val>

This command enables bits in the Standard Event Enable register. Bits enabled and set in this register will set the Standard Event Status Summary bit (bit 5) in the Status Byte register. When bit 5 (decimal 32) in the Status Byte register is set, you can read the Standard Event register using the \*ESR command and determine the cause.

The Standard Event Enable register state (bits enabled with this command) is not affected by signal generator preset or \*RST. The register will be cleared when the signal generator is turned off unless the command \*PSC is used before turning it off.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### Example

\*ESE 129

This command enables bit 0 (decimal 1, Operation Complete) and bit 7 (decimal 128, Power On) in the Standard Event Status Enable register.

**Range**                0-255

### \*ESE?

**Supported**            All Models

\*ESE?

This query returns the decimal sum of the enabled bits in the Standard Event Enable register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

## \*ESR?

**Supported**            All Models

---

**NOTE**    This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared. Refer to the *Agilent Signal Generators Programming Guide* for more information.

---

\*ESR?

This query returns the decimal sum of the bits set in the Standard Event register.

## \*IDN?

**Supported**            All Models

\*IDN?

This query requests an identification string from the signal generator. The IDN string consists of the following information:

<company\_name>, <model\_number>, <serial\_number>, <firmware\_revision>

The identification information can be modified. Refer to [:IDN](#) command for more information.

**Key Entry**            **Diagnostic Info**

## \*OPC

**Supported**            All Models

\*OPC

The Operation Complete (OPC) command sets bit 0 in the Standard Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

## \*OPC?

**Supported**            All Models

\*OPC?

The Operation Complete (OPC) query returns the ASCII character 1 in the Standard Event register indicating completion of all pending operations.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

## \*PSC

**Supported** All Models

\*PSC ON|OFF|1|0

The power-on Status Clear (PSC) command controls the automatic power-on clearing of the Service Request Enable register, the Standard Event Status Enable register, and the device-specific event enable registers.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

ON (1) This choice enables the power-on clearing of the listed registers.

OFF (0) This choice disables the clearing of the listed registers and they retain their status when a power-on condition occurs.

### Example

```
*PSC ON
```

This command clears all listed registers at power-on.

## \*PSC?

**Supported** All Models

```
*PSC?
```

The power-on Status Clear (PSC) query returns the flag (1 or 0) setting as enabled by the \*PSC command.

## \*RCL

**Supported** All Models

```
*RCL <reg> , <seq>
```

The Recall (RCL) command recalls the state from the specified memory register <reg> in the specified sequence <seq>.

**Range** registers: 0–99 Sequences: 0–9

**Key Entry** **RECALL Reg** **Select Seq:**

## \*RST

**Supported** All Models

```
*RST
```

The Reset (RST) command resets most signal generator functions to a factory-defined state.

Each command description in this reference shows the \*RST value if the signal generator's setting is affected.

## \*SAV

**Supported** All Models

\*SAV <reg> , <seq>

The Save (SAV) command saves the state of the signal generator to the specified memory register <reg> of the specified sequence <seq>. Settings such as frequency, attenuation, power, and settings that do not survive a power cycle or an instrument reset can be saved. Data formats, arb setups, list sweep values, table entries, and so forth are not stored. Only a reference to the data file name is saved. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* and *Agilent Signal Generators Programming Guide* for more information on saving and recalling instrument states.

**Range** registers: 0–99 Sequences: 0–9

**Key Entry** Save Reg Save Seq[n] Reg[nn]

## \*SRE

**Supported** All Models

\*SRE <val>

The Service Request Enable (SRE) command enables bits in the Service Request Enable register. Bits enabled and set in this register will set bits in the Status Byte register.

The variable <val> is the decimal sum of the bits that are enabled. Bit 6 (value 64) is not available in this register and therefore cannot be enabled by this command. Because bit 6 is not available, entering values from 64 to 127 is equivalent to entering values from 0 to 63.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

The setting enabled by this command is persistent, in that it is not affected by cycling the signal generator power, preset or the \*RST command.

**Range** 0 to 63, 128 to 191

## \*SRE?

**Supported** All Models

\*SRE?

The Service Request Enable (SRE) query returns the decimal sum of bits enabled in the Service Request Enable register. Bit 6 (decimal 64) is not available in this register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 63, 128 to 191

### **\*STB?**

**Supported**            All Models

\*STB?

This command reads the decimal sum of the bits set in the Status Byte register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range**                0 to 255

### **\*TRG**

**Supported**            All Models

\*TRG

The Trigger (TRG) command triggers the device if BUS is the selected trigger source, otherwise, \*TRG is ignored. For more information on triggers; refer to “:TRIGger[:SEquence]:SOURce” on page 106.

### **\*TST?**

**Supported**            All Models

\*TST?

The Self-Test (TST) query initiates the internal self-test and returns one of the following results:

- 0                        This shows that all tests passed.
- 1                        This shows that one or more tests failed.

**Key Entry**            **Run Complete Self Test**

### **\*WAI**

**Supported**            All Models

\*WAI

The Wait-to-Continue (WAI) command causes the signal generator to wait until all pending commands are completed, before executing any other commands.

## Low-Band Filter Subsystem

### **[[:SOURce]:LBFilter**

**Supported** All Models with Option 1EH or 521

[[:SOURce]:LBFilter ON|OFF|1|0

[[:SOURce]:LBFilter?

This command enables or disables the low-band filter located in the RF path. Use this filter to reduce harmonics below 2 GHz.

**\*RST** 0

**Key Entry** Low Pass Filter below 2 GHz Off On

## Memory Subsystem (:MEMory)

### **:CATalog:BINary**

**Supported** All Models

:MEMory:CATalog:BINary?

This command outputs a list of binary files. The return data will be in the following form:

<mem\_used>, <mem\_free>{, "<file\_listing>"}

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

"<file\_name,file\_type,file\_size>"

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** Binary

### **:CATalog:BIT**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:BIT?

This command outputs a list of bit files. The return data will be in the following form:

<mem\_used>, <mem\_free>{, "<file\_listing>"}

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

"<file\_name,file\_type,file\_size>"

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** Bit

## :CATalog:DMOD

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:DMOD?

This command outputs a list of arbitrary waveform digital modulation files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **DMOD**

## :CATalog:FIR

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:FIR?

This command outputs a list of finite impulse response (FIR) filter files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **FIR**

## :CATalog:FSK

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:FSK?

This command outputs a list of frequency shift keying (FSK) files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **FSK**



## **:CATalog:IQ**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:IQ?

This command outputs a list of IQ files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** I/Q

## **:CATalog:LIST**

**Supported** All Models

:MEMory:CATalog:LIST?

This command outputs a list of List Sweep files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** List

## **:CATalog:MMod**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:MMod?

This command outputs a list of arbitrary waveform multicarrier digital modulation (MMod) files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** MMOD

## **:CATalog:MTONE**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:MTONE?

This command outputs a list of arbitrary waveform multitone files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **MTONE**

## **:CATalog:SEQ**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:SEQ?

This command outputs a list of arbitrary waveform sequence files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **Seq**

## **:CATalog:SHAPE**

**Supported** E8267D with Option 601 or 602

:MEMory:CATalog:SHAPE?

This command outputs a list of burst shape files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry** **Shape**

## **:CATalog:STATe**

**Supported**            All Models

:MEMory:CATalog:STATe?

This command outputs a list of state files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry**            **State**

## **:CATalog:UFLT**

**Supported**            All Models

:MEMory:CATalog:UFLT?

This command outputs a list of user-flatness correction files. The return data will be in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [File Name Variables](#) for information on the file name syntax.

**Key Entry**            **User Flatness**

## **:CATalog[:ALL]**

**Supported**            All Models

:MEMory:CATalog[:ALL]?

This command outputs a list of all files in the memory subsystem, but does not include files stored in the Option 601 or 602 baseband generator memory. The return data is in the following form:

```
<mem_used>,<mem_free>{,"<file_listing>"}
```

The signal generator returns the two memory usage parameters and as many file listings as there are files in the memory subsystem. Each file listing parameter is in the following form:

```
"<file_name,file_type,file_size>"
```

Refer to [Table 2-1](#) for file types, and to [File Name Variables](#) for file name syntax.

**Key Entry**            **All**

## :COPY[:NAME]

**Supported**            All Models

```
:MEMory:COPY[:NAME] "<src_name>" , "<dest_name>"
```

This command copies the data from one file into another file. The file can use the same name if the specified directory is different. For example, if the file resides in non-volatile waveform memory (NVWFM) it can be copied, using the same name, to the signal generator's volatile memory (WFM1).

When copying a waveform or marker file from volatile to non-volatile memory, the associated marker or waveform file is also copied.

"<src\_name>"            This variable names a file residing in memory that will be copied. For information on the file name syntax, refer to ["File Name Variables" on page 10](#).

"<dest\_name>"           This variable names the file that is a copy of the "<src\_name>" file.

### Example

```
:MEM:COPY "/USER/IQ/4QAM" , "/USER/IQ/test_QAM"
```

The preceding example copies the 4QAM file in the signal generator's /USER/IQ directory to a file named test\_QAM and saves it in the same directory.

**Key Entry**            **Copy File**

## :DATA

**Supported**            All Models

```
:MEMory:DATA "<file_name>" , <data_block>
```

```
:MEMory:DATA? "<file_name>"
```

This command loads waveform data into signal generator memory using the <data\_block> parameter and saves the data to a file designated by the "<file\_name>" variable. The query returns the file contents of the file as a datablock.

The waveform file must be located in volatile waveform memory (WFM1) before it can be played by the signal generator's Dual ARB player. For downloads directly into volatile waveform memory use the path "WFM1:<file\_name>". For downloads to non-volatile waveform memory, use the path "NVWFM:<file\_name>".

Refer to [File Name Variables](#) for information on the file name syntax.

"<file\_name>"           This variable names the destination file, including the directory path. Refer to [ARB Waveform File Directories](#) for information on directory paths and the file name syntax.

<data\_block>           This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**NOTE** ARB waveform files created using the :DATA command cannot be retrieved or uploaded. Attempting to do so will cause the signal generator to display the message: ERROR:221, Access denied. To download ARB data to files for later retrieval, use the :DATA:UNPRotected command on [page 53](#).

### Example

```
:MEM:DATA "NVWFM:IQ_Data",#210Qaz37pY9oL
```

The preceding example downloads 10 bytes of data to a file, IQ\_Data., in the signal generator's non-volatile memory. The table shown below describes the command parameters.

- "NVWFM:IQ\_Data"                    **IQ\_Data is the data filename. The directory path is specified along with the filename**
- #210Qaz37pY9oL                    Data block
  - #                                    This character indicates the beginning of the data block
  - 2                                    Number of digits in the byte count
  - 10                                    Byte count
  - Qaz37pY9oL                        10 bytes of data

### :DATA:APPend

**Supported**                    All Models

```
:MEMory:DATA:APPend "<file_name>",<data_block>
```

This commands appends data to an existing file stored in signal generator memory.

"<file\_name>"                    This variable names the destination file and directory path. Refer to [File Name Variables](#) for information on the file name syntax.

<data\_block>                    This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable. The file length parameters are used by the signal generator for allocating memory.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:APPend "NVWFM:IQ_Data",#14Y9oL
```

The preceding example downloads and appends the data, Y9oL, to an existing file named IQ\_Data stored in the signal generator's non-volatile memory (NVWFM).

- "NVWFM:IQ\_Data"                    **IQ\_Data is the filename to append data to. The directory path is specified along with the filename.**
- #14Y9oL                            Data block
  - #                                    This character indicates the beginning of the data block

- "NVWFM:IQ\_Data"                    **IQ\_Data is the filename to append data to. The directory path is specified along with the filename.**
  - 1                                        Number of digits in the byte count
  - 4                                        Byte count
  - Y9oL                                    4 bytes of data

## :DATA:BIT

Supported                    E8267D with Option 601 or 602

```
:MEMory:DATA:BIT "<file_name>",<bit_count>,<data_block>
```

```
:MEMory:DATA:BIT? "<file_name>"
```

This command loads bit data into signal generator memory using the <bit\_count> and <data\_block> parameters and saves the data to a file designated by the "<file\_name>" variable. The query returns the bit count, file length information, and the data.

"<file\_name>"                This variable names the destination file and the directory path. Refer to [File Name Variables](#) for information on the file name syntax.

<bit\_count>                    This number represents the number of bits in the data block.

<data\_block>                 This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable. The file length parameters are used by the signal generator for allocating memory.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:BIT "/USER/BIT/Test_Data",16,#12Qz
```

The preceding example downloads bit data to the file, Test\_Data. The table below describes the command parameters.

- "/USER/BIT/Test\_Data"                **Test\_Data is the bit data filename. The directory path is specified along with the filename**
- 16                                        Number of bits in the data block
- #12Qz                                    Data block
  - #                                        This character indicates the beginning of the data block
  - 1                                        Number of digits in the byte count
  - 2                                        Byte count
  - Qz                                        16 bits of data (ascii representation of bit data)

## :DATA:FIR

**Supported** E8267D with Option 601 or 602

```
:MEMory:DATA:FIR "<file_name>",<osr>,<coefficient>{,<coefficient>}
:MEMory:DATA:FIR? "<file_name>"
```

This command loads oversample ratio (OSR) and user-defined finite impulse response (FIR) coefficient data into a file in the signal generator's non-volatile memory (NVWFM). The query returns the oversample ratio and coefficient data.

"<file\_name>" This variable is the directory path and file name of the destination file. Refer to [File Name Variables](#) for information on the file name syntax.

osr The OSR is the number of filter taps per symbol.

coefficient This variable is the FIR coefficient. The maximum number of coefficients is 1024.

{,<coefficient>} This optional variable is used when you enter additional coefficients.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:FIR "/USER/FIR/FIR_1",4,0,0,0,0,0,0.000001,0.000012,0.000132,
0.001101,0.006743,0.030588,0.103676,0.265790,0.523849,0.809508,1,1,
0.809508,0.523849,0.265790,0.103676,0.030588,0.006743,0.001101,0.000132,0.000012,0.00
0001,0,0,0,0,0
```

The preceding example downloads FIR coefficient and oversampling ratio data to the signal generator's non-volatile memory in a file named FIR\_1.

**Range** *osr:* 1–32  
*coefficient:* –1000 to 1000

**Key Entry** **Oversample Ratio**

## :DATA:FSK

**Supported** E8267D with Option 601 or 602

```
:MEMory:DATA:FSK "<file_name>",<num_states>,<f0>,<f1>,...<f(n)>
[,<diff_state>,<num_diff_states>,<diff1>,...<diff(n)>]
:MEMory:DATA:FSK? "<file_name>"
```

This command loads custom frequency shift keying (FSK) data into a file in the signal generator's non-volatile memory (NVWFM).

The query returns data in the following form:

```
<num_states>,<f0>,<f1>,...<f(n)>,<diff_state>,<num_diff_states>,<diff1>,...<diff(n)>
"<file_name>" This variable string identifies the name of the FSK file. The filename must be enclosed with quotation marks. Refer to File Name Variables for information on the file name syntax.
```

- <num\_states> This variable identifies the number of frequency states.
- <f0> This variable identifies the value of the first frequency state.
- <f1>,...<f(n)> This variable identifies the value of the second and subsequent frequency states with a frequency resolution of 0.1Hz.
- <diff\_state> This variable enables or disables differential encoding.
- <num\_diff\_states> This variable identifies the number of differential states.
- <diff0> This variable identifies the value of the first differential state.
- <diff1>,...<diff(n)> This variable identifies the value of the second and subsequent differential states.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

**Example**

```
:MEM:DATA:FSK "/USER/FSK/4FSK",4,-2kHz,-1kHz,2kHz,1kHz,ON,2,1,0
```

The preceding example downloads a four-level FSK data to a file named 4FSK that has four states (frequencies): -2kHz, -1kHz, 2kHz, 1kHz; differential encoding is toggled ON, and there are two differential states 1 and 0. The table shown below describes the command parameters.

- **"/USER/FSK/4FSK"** **4FSK is the FSK data filename. The directory path is specified along with the filename**
- 4 Number of states
- 2kHz First frequency state
- 1kHz Second frequency state
- 2kHz Third frequency state
- 1kHz Fourth frequency state
- ON Differential encoding is on
- 2 Number of differential states
- 1 Value of the first differential state.
- 0 Value of the second differential state.

**Range** *num\_diff\_states:* 0-256  
*num\_states:* 2-16  
*f0-f(n):* -20MHz to 20MHz  
*diff0-diff(n):* -128 to 127



## :DATA:IQ

**Supported** E8267D with Option 601 or 602

```
:MEMory:DATA:IQ "<file_name>",<offsetQ>,<num_states>,<i0>,<q0>,<i1>,<q1>,...<i(n)>,<q(n)>[,<diff_state>,<num_diff_states>,<diff0>,<diff1>,...<diff(n)>]
:MEMory:DATA:IQ? "<file_name>"
```

This command loads custom I/Q data into a file in the signal generator's non-volatile waveform memory (NVWFM).

The query returns data in the following form:

```
<offsetQ>,<num_states>,<i0>,<q0>,<i1>,<q1>,...<i(n)>,<q(n)>,<diff_state>,<num_diff_states>,<diff0>,<diff1>,...<diff(n)>
```

"<file_name>"	This variable string identifies the name of the I/Q file. The filename must be enclosed with quotation marks. Refer to <a href="#">File Name Variables</a> for information on the file name syntax.
<offsetQ>	This variable enables (1) or disables (0) the Q output delay by 1/2 symbol from the I output.
<num_states>	This is the number of symbols.
<i0>...<i(n)>	This is the I value of the first and subsequent I symbols.
<q0>...<q(n)>	This is the Q value of the first and subsequent Q symbols.
<diff_state>	This variable enables and disables differential encoding.
<num_diff_states>	This variable identifies the number of differential states.
<diff0>	This variable identifies the value of the first differential state.
<diff1...diff(n)>	This variable identifies the value of the second and subsequent differential states.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:IQ "/USER/IQ/Test_BPSK",1,2,1,0,0,0
```

The preceding example loads and stores a two-symbol I/Q file named Test\_BPSK that has a Q offset. The table shown below describes the command parameters.

- "/USER/IQ/Test\_BPSK" **Test\_Data is the bit data filename. The directory path is specified along with the filename**
- 1 Q Offset. The Q output delay is enabled.
- 2 Number of symbols
- 1 Value of the first I symbol
- 0 Value of the first Q symbol.
- 0 Value of the second I symbol
- 0 Value of the second Q symbol.

**Range**                    num\_states: 2–256  
                          *i0–i(n)*: –1 to 1  
                          *q0–q(n)*: –1 to 1  
                          num\_diff\_states: 0–256  
                          *diff0–diff(n)*: –128 to 127

## **:DATA:PRAM:FILE:BLOCK**

**Supported**                E8267D with Option 601 or 602

:MEMory:DATA:PRAM:FILE:BLOCK "<file\_name>", <data\_block>

This command loads block-formatted data directly into pattern RAM volatile memory (WFM1). Pattern RAM memory describes how memory (WFM1) is used and is not a distinct piece of memory. A PRAM file is specified as an array of bytes. No directory path name is needed.

"<file\_name>"              This variable names the destination file. Refer to [File Name Variables](#) for information on the file name syntax.

<data\_block>                This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable. The file length parameters are used by the signal generator for allocating memory.

Pattern Ram files are binary files downloaded directly into waveform memory as an array of bytes. Each byte specifies a data bit (LSB 0), a burst bit (BIT 2), and an Event 1 output bit (BIT 6). Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### **Example**

:MEM:DATA:PRAM:FILE:BLOCK "PRAM\_Data",#14Yq8L

The preceding example downloads PRAM data to a file named PRAM\_Data into the signal generator's volatile memory.

- "PRAM\_Data"                    **PRAM\_Data is the data filename. PRAM files are saved to the signal generator's non-volatile memory**
- #14Yq8L                        Data block
  - #                                This character indicates the beginning of the data block
  - 1                                Number of digits in the byte count
  - 4                                Byte count
  - Yq8L                            4 bytes of data

## :DATA:PRAM:FILE:LIST

**Supported** E8267D with Option 601 or 602

```
:MEMory:DATA:PRAM:FILE:LIST "<file_name>" ,<uint8>[ ,<uint8>,<...>]
```

This command loads list-formatted data directly into pattern RAM volatile memory (WFM1). Pattern RAM memory describes how memory (WFM1) is used and is not a distinct piece of memory. A PRAM file is specified as an array of bytes.

---

**NOTE** This command should be preceded by a \*WAI (Wait-to-Continue) command to ensure that all pending operations are completed, before loading the list.

---

"<file_name>"	This variable names the destination file. Refer to <a href="#">File Name Variables</a> for information on the file name syntax.
<uint8>	This variable is any of the valid 8-bit, unsigned integer values between 0 and 255.
[,<uint8>,<...>]	This variable identifies the value of the second and subsequent 8-bit unsigned integer variables.

Pattern Ram files are binary files downloaded directly into waveform memory as an array of bytes. Each byte specifies a data bit (LSB 0), a burst bit (BIT 2), and an Event 1 output bit (BIT 6). Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:PRAM:LIST "Pram_Data" , 85,21,21,20,20,100
```

The preceding example downloads PRAM data, in list format, to a file named Pram\_Data in the signal generator's volatile memory (WFM1).

- "Pram\_Data" **Pram\_Data is the data filename. PRAM files are saved to the signal generator's non-volatile memory**
- 85 The first 8-bit integer value
- 21,21,20,20,100 Subsequent 8-bit integer values.

**Range** 0 to 255

## :DATA:PRAM?

---

**NOTE** This query is no longer supported; however, it is still valid for backward compatibility. Refer to [":DATA:PRAM?" on page 348](#) for information on this command.

---

## :DATA:PRAM:BLOCK

---

**NOTE** This command has been replaced by “:DATA:PRAM:FILE:BLOCK” on page 50. This command is no longer supported; however, it is still valid for backward compatibility. Refer to “:DATA:PRAM:BLOCK” on page 348 for information.

---

## :DATA:PRAM:LIST

---

**NOTE** This command has been replaced by “:DATA:PRAM:FILE:LIST” on page 51. This command is no longer supported; however, it is still valid for backward compatibility. Refer to “:DATA:PRAM:LIST” on page 348 for information.

---

## :DATA:SHAPE

**Supported** E8267D with Option 601 or 602

```
:MEMory:DATA:SHAPE "<file_name>",<rise_pnts>,<rp0>,<rp1>,...<fall_points>,<fp0>,  
<fp1>,...<fp(n)>  
:MEMory:DATA:SHAPE? "<file_name>"
```

This command loads a burst shape file into the signal generator’s non-volatile memory (NVWFM).

"<file\_name>" This variable names the destination file and directory path. Refer to [File Name Variables](#) for information on the file name syntax.

rise\_pnts This variable indicates the number of rise points used to describe the burst shape rising slope.

rp0,...rp(n) This variable defines each successive rise point, where 0 is no power and 1 is full power.

fall\_points This variable indicates the number of fall points used to describe the burst shape falling slope.

fp0,...fp(n) This variable defines each successive fall point, where 1 is full power and 0 is no power.

Refer the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:SHAP "/USER/SHAPE/Shape_File",6,0,0.2,0.4,0.6,0.8,1.0,2,0.5,0
```

The preceding example loads shape data to a file named Shape\_File in the signal generator’s non-volatile memory.

- "/USER/SHAPE/Shape\_File" **Shape\_File is the data filename. The directory path is specified along with the file name.**
- 6 Number of rise points describing the burst shape.

- `"/USER/SHAPE/Shape_File"`      **Shape\_File is the data filename. The directory path is specified along with the file name.**
- 0,0.2,0.4,0.6,0.8,1.0              Rise point values.
- 2    Number of fall points describing the burst shape.
- 0.5,0                                    Fall point values.

**Range**                      *num\_rise\_points:* 2-256  
                                   *num\_fall\_points:* 2-256  
                                   *rp0-rp(n):* 0.0-1.0  
                                   *fp0-fp(n):* 0.0-1.0

### **:DATA:UNPRotected**

**Supported**                E8267D with Option 601 or 602

:MEMory:DATA:UNPRotected "<file\_name>",<data\_block>

This command allows you to download data and store it in a file on the signal generator with the ability to retrieve it. This command is intended for downloading waveform data; however, you can use it to download all types of data.

---

**NOTE**    If you do not use the UNPRotected command when downloading a waveform file, you will not be able to retrieve or upload the file. Attempting to do so will cause the signal generator to display the message: ERROR:221, Access denied.

---

The UNPRotected command does not require the directory path in the "<file\_name>" parameter if the destination directory is BINARY.

Waveform files created with Agilent's Signal Studio are encrypted. These files can be used in other signal generators (provided the other signal generator has the same options and licenses required by the file) only if the SECUREWAVE directory path is specified in both the download and upload command parameters. The securewave directory path is SNVWFM: for non-volatile waveform memory and SWFM1: for volatile waveform memory.

"<file\_name>"              This variable names the destination file and directory path. Refer to [File Name Variables](#) for information on the file name syntax.

<data\_block>                This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

### Example

```
:MEM:DATA:UNPR "NVWFM:Data_File",#18Qz37pY9o
```

The preceding example downloads waveform data to a file named Data\_File in the signal generator's non-volatile memory. The table shown below describes the command parameters.

- "NVWFM:Data\_File"                      **Data\_File is the data filename. The directory path is implied along with the filename.**
- #18Qz37pY9o                              Data block
  - #    This character indicates the beginning of the data block
  - 1    Number of digits in the byte count
  - 8    Byte count
  - Qz37pY9o                                8 bytes of data

### :DElete:ALL

**Supported**                      All Models

---

**CAUTION**                      Using this command deletes all user files including binary, list, state, and flatness correction files, and any saved setups that use the front panel table editor. However, this does not include files stored in the Option 601 or 602 baseband generator memory. You cannot recover the files after executing this command.

---

```
:MEMory:DElete:ALL
```

This command clears the file system of all user files.

**Key Entry**                      **Delete All Files**

### :DElete:BINary

**Supported**                      All Models

```
:MEMory:DElete:BINary
```

This command deletes all binary files.

**Key Entry**                      **Delete All Binary Files**

### :DElete:BIT

**Supported**                      E8267D with Option 601 or 602

```
:MEMory:DElete:BIT
```

This command deletes all bit files.

**Key Entry**                      **Delete All Bit Files**

### **:DElete:DMOD**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:DMOD

This command deletes all arbitrary waveform digital modulation (DMOD) files.

**Key Entry** **Delete All ARB DMOD Files**

### **:DElete:FIR**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:FIR

This command deletes all finite impulse response (FIR) filter files.

**Key Entry** **Delete All FIR Files**

### **:DElete:FSK**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:FSK

This command deletes all frequency shift keying (FSK) files.

**Key Entry** **Delete All FSK Files**

### **:DElete:IQ**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:IQ

This command deletes all I/Q files.

**Key Entry** **Delete All I/Q Files**

### **:DElete:LIST**

**Supported** All Models

:MEMory:DELeTe:LIST

This command deletes all List files.

**Key Entry** **Delete All List Files**

### **:DElete:MDMod**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:MDMod

This command deletes all arbitrary waveform multicarrier digital modulation (MDMod) files.

**Key Entry** **Delete All ARB MDMOD Files**

### **:DElete:MTONe**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:MTONe

This command deletes all arbitrary waveform multitone files.

**Key Entry** **Delete All ARB MTONE Files**

### **:DElete:SEQ**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:SEQ

This command deletes all sequence files.

**Key Entry** **Delete All Sequence Files**

### **:DElete:SHAPE**

**Supported** E8267D with Option 601 or 602

:MEMory:DELeTe:SHAPE

This command deletes all burst shape files.

**Key Entry** **Delete All Shape Files**

### **:DElete:STATe**

**Supported** All Models

:MEMory:DELeTe:STATe

This command deletes all state files.

**Key Entry** **Delete All State Files**



## **:DElete:UFLT**

**Supported**            All Models

:MEMory:DElete:UFLT

This command deletes all user-flatness correction files.

**Key Entry**            **Delete All UFLT Files**

## **:DElete[:NAME]**

**Supported**            All Models

:MEMory:DElete[:NAME] "<file\_name>"

This clears the user file system of "<file\_name>". When deleting an ARB waveform file, the associated marker and header files are also deleted.

Refer to [File Name Variables](#) for information on the file name syntax.

### **Example**

```
:MEM:DEL "/USER/WAVEFORM/Test_Data"
```

The preceding example deletes the file named Test\_Data from the signal generator's non-volatile memory.

**Key Entry**            **Delete File**

## **:FREE[:ALL]**

**Supported**            All Models

:MEMory:FREE[:ALL]?

This command returns the number of bytes left in the user file system.

**Key Entry**            **All**

## **:LOAD:LIST**

**Supported**            All Models

:MEMory:LOAD:LIST "<file\_name>"

This command loads a List Sweep file.

### **Example**

```
:MEM:LOAD:LIST "List_Data"
```

The preceding example loads the file "List\_Data" into volatile waveform memory.

**Key Entry**            **Load From Selected File**

## **:MOVE**

**Supported**            All Models

```
:MEMory:MOVE "<src_file>","<dest_file>"
```

This command renames the `src_file` to `dest_file` in the signal generator's memory catalog.

Refer to [File Name Variables](#) for information on the file name syntax.

### **Example**

```
:MEM:MOV "NVWFM:Test_Data","NVWFM:New_Data"
```

The preceding example renames the file `Test_Data` to `New_Data` in the signal generator's non-volatile memory directory.

**Key Entry**            **Rename File**

## **:STATe:COMMeNt**

**Supported**            All Models

```
:MEMory:STATe:COMMeNt <reg_num>,<seq_num>,"<comment>"
```

```
:MEMory:STATe:COMMeNt? <reg_num>,<seq_num>
```

This command lets you to add a descriptive comment to the saved instrument in the state register, `<reg_num>,<seq_num>`. Comments can be up to 55 characters long.

### **Example**

```
:MEM:STAT:COMM 00,1,"ARB file using external reference"
```

The preceding example writes a descriptive comment to the state file saved in register 00, sequence 1.

**Key Entry**            **Add Comment To Seq[n] Reg[nn]**

## **:STORe:LIST**

**Supported**            All Models

```
:MEMory:STORe:LIST "<file_name>"
```

This command stores the current list sweep data to a file.

Refer to [File Name Variables](#) for information on the file name syntax.

### **Example**

```
:MEM:STOR:LIST "Test_Data"
```

The preceding example writes list sweep data to a file named `Test_Data` and stores the file in the signal generator's non-volatile memory, List directory.

**Key Entry**            **Store To File**

## Mass Memory Subsystem (:MMEMory)

### :CATalog

**Supported**            All Models

```
:MMEMory:CATalog? "<msus>"
```

This command outputs a list of the files from the specified file system. The variable "<msus>" (mass storage unit specifier) represents a file system. The file systems and types are shown in [Table 2-1](#).

**Table 2-1**

File System	File Type
BIN - Binary file	BIN
BIT	BIT
DMOD - ARB digital modulation file	DMOD
FIR - finite impulse response filter file	FIR
FSK - frequency shift keying modulation file	FSK
I/Q - modulation file	IQ
LIST - sweep list file	LIST
MDMOD - ARB multicarrier digital modulation file	MDM
MTONE - ARB multitone file	MTON
NVMKR - non-volatile arbitrary waveform marker file	NVMKR
NVWFM - non-volatile arbitrary waveform file	NVWFM
SEQ - ARB sequence file	SEQ
SHAPE - burst shape file	SHAP
STATE	STAT
USERFLAT - user-flatness file	UFLT
WFM1 - waveform file	WFM1

The return data will be in the following form:

```
<mem_used>, <mem_free>{, "<file_listing>"}
```

The signal generator will return the two memory usage parameters and as many file listings as there are files in the specified file system. Each file listing will be in the following format:

```
"<file_name, file_type, file_size>"
```

Refer to [MSUS \(Mass Storage Unit Specifier\) Variable](#) for information on the use of the "<msus>" variable.

<b>Key Entry</b>	<b>Bit</b>	<b>I/Q</b>	<b>Shape</b>	<b>MTONE</b>	<b>DMOD</b>
	<b>Seq</b>	<b>List</b>	<b>Binary</b>	<b>NVMKR</b>	<b>NVWFM</b>

**FIR**      **State**      **WFM1**      **MDMOD**      **User Flatness**  
**FSK**

## :COPY

**Supported**      All Models

```
:MMEMory:COPY[:NAME] "<src_name>" "<dest_name>"
```

This command copies the data from one file into another file. The file can use the same name if the specified directory is different. For example, if the file resides in non-volatile waveform memory (NVWFM) it can be copied, using the same name, to the signal generator's volatile memory (WFM1)

"<src\_name>"      This variable names a file residing in memory that will be copied. For information on the file name syntax, see [File Name Variables](#).

"<dest\_name>"      This variable names the file that is a copy of the "<src\_name>" file.

### Example

```
:MMEM:COPY "/USER/IQ/4QAM" "/USER/IQ/test_QAM"
```

The preceding example copies the 4QAM file in the signal generator's /USER/IQ directory to a file named test\_QAM and saves it in the same directory.

**Key Entry**      **Copy File**

## :DATA

**Supported**      All Models

```
:MMEMory:DATA "<file_name>" ,<data_block>
```

```
:MMEMory:DATA? "<file_name>"
```

This command loads waveform data into signal generator memory using the <data\_block> parameter and saves the data to a file designated by the "<file\_name>" variable. The query returns the file contents of the file as a datablock.

The waveform file must be located in volatile waveform memory (WFM1) before it can be played by the signal generator's Dual ARB player. For downloads directly into volatile waveform memory use the path "WFM1:<file\_name>". For downloads to non-volatile waveform memory, use the path "NVWFM:<file\_name>".

Refer to [File Name Variables](#) for information on the file name syntax.

"<file\_name>"      This variable names the destination file, including the directory path. Refer to [ARB Waveform File Directories](#) for information on directory paths and the file name syntax.

<data\_block>      This parameter represents the data and file length parameters. The data in the file is represented by the <data\_block> variable. The file length parameters are used by the signal generator for allocating memory.

Refer to the *Agilent Signal Generators Programming Guide* for more information on downloading and using files.

---

**NOTE** Files created using the :DATA command cannot be retrieved or uploaded. Attempting to do so will cause the signal generator to display the message: ERROR:221, Access denied. To download data to files for later retrieval, use the :DATA:UNPRotected command on [page 53](#).

---

**Example**

```
:MMEM:DATA "NVWFM:IQ_Data",#210Qaz37pY9oL
```

The preceding example downloads 10 bytes of data to a file, IQ\_Data., in the signal generator's non-volatile memory. The table shown below describes the command parameters.

- "NVWFM:IQ\_Data"                    **IQ\_Data is the data filename. The directory path is specified along with the filename**
- #210Qaz37pY9oL                    Data block
  - #                                    This character indicates the beginning of the data block
  - 2                                    Number of digits in the byte count
  - 10                                    Byte count
  - Qaz37pY9oL                        10 bytes of data

**:DElete:NVWFm**

**Supported**                    E8267D with Option 601 or 602

```
:MMEMory:DElete:NVWFm
```

This command clears the memory file system of all non-volatile arbitrary waveform (NVWFM) files.

**Key Entry**                    **Delete All NVWFM Files**

**:DElete:WFM**

**Supported**                    E8267D with Option 601 or 602

```
:MMEMory:DElete:WFM
```

This command clears the memory file system of all volatile arbitrary waveform files. For backwards compatible command, refer to "[:DElete:WFM1](#)" on [page 348](#).

**Key Entry**                    **Delete All WFM Files**

**:DElete[:NAME]**

**Supported**                    All Models

```
:MMEMory:DElete[:NAME] "<file_name>",<msus>"]
```

This command clears the memory file system of "<file\_name>" with the option of specifying the file system [<msus>"] separately.

The variable "<msus>" (mass storage unit specifier) represents the file system. For a list of the file systems refer to [Table 2-1](#). For information on the mass storage unit specifier, refer to "[MSUS \(Mass](#)

Storage Unit Specifier) Variable” on page 11.

If the optional variable "<msus>" is omitted, the file name needs to include the file system extension. Refer to [File Name Variables](#) for information on the file name syntax.

**Example**

```
:MMEM:DEL "/USER/BIN/Test_Data"  
:MMEM:DEL "Test_Data", ":BIN"
```

The preceding examples delete the file named Test\_Data from the signal generator’s USER/BIN directory. The first example uses the full file name path while the second example uses the "<msus>" specifier.

**Key Entry**            **Delete File**

**:HEADer:CLEAr**

**Supported**            E8267D

```
:MMEMory:HEADer:CLEAr "<file_name>"
```

This command deletes header file information for the waveform file "<file\_name>". This command does not require a personality modulation to be on. The header file contains signal generator settings and marker routings associated with the waveform file. Refer to [File Name Variables](#) for information on the file name syntax.

**Example**

```
:MMEM:HEAD "/USER/WAVEFORM/Test_Data"
```

The preceding example deletes header file information for the Test\_Data waveform file.

**Key Entry**            **Clear Header**

**:HEADer:DESCRiption**

**Supported**            E8267D

```
:MMEMory:HEADer:DESCRiption "<file_name>","<description>"  
:MMEMory:HEADer:DESCRiption? "<file_name>"
```

This command inserts a description for the header file named. The header description is limited to 32 characters.

Refer to [File Name Variables](#) for information on the file name syntax.

**Example**

```
:MMEM:HEAD:DESC "/USER/WAVEFORM/Test_Data","This is new header data"
```

The preceding example inserts a description into the Test\_Data header file. In this example, the file is located in the signal generator’s non-volatile waveform memory.

**\*RST**                    N/A

**Key Entry**            **Edit Description**

## **:HEADer:ID?**

**Supported** All Models

MMEMory:HEADer:ID?

This query returns a unique waveform identification.

Refer to [File Name Variables](#) for information on the file name syntax.

## **:LOAD:LIST**

**Supported** All Models

:MMEMory:LOAD:LIST "<file\_name>"

This command loads list data from the List file "<file\_name>".

Refer to [File Name Variables](#) for information on the file name syntax.

### **Example**

```
:MMEM:LOAD:LIST "Sweep_Data"
```

The preceding example loads sweep configuration data from the Sweep\_Data List file.

**Key Entry** Load From Selected File

## **:MOVE**

**Supported** All Models

:MMEMory:MOVE "<src\_file>", "<src\_file\_1>"

This command renames the src\_file to src\_file\_1 in the signal generator's memory catalog.

Refer to [File Name Variables](#) for information on the file name syntax and using quotes for different programming languages.

### **Example**

```
:MMEM:MOV "NVWFM:Test_Data", "NVWFM:New_Data"
```

The preceding example renames the file Test\_Data to New\_Data located in non-volatile memory (NVWFM).

**Key Entry** Rename File

## **:STORe:LIST**

**Supported** All Models

:MMEMory:STORe:LIST "<file\_name>"

This command copies the current list sweep data to the "<file\_name>" and saves it in the catalog of List files.

Refer to [File Name Variables](#) for information on the file name syntax.

### Example

```
:MMEM:STOR:LIST "Sweep_Data"
```

The preceding example stores the current list sweep data to the file Sweep\_Data in the signal generator's catalog of List files.

**Key Entry**            **Store To File**



## Output Subsystem (:OUTPut)

### :BLANking:AUTO

**Supported** All Models

```
[ :SOURce]:OUTPut:BLANking:AUTO ON|OFF|1|0
[ :SOURce]:OUTPut:BLANking:AUTO?
```

This command sets the state for automatic RF Output blanking. Blanking occurs when the RF output is momentarily turned off as the sweep transitions from one frequency segment (band) to another, allowing the signal to settle. Blanking also occurs during the retrace, so the signal can settle before the next sweep. In CW mode, blanking occurs whenever you change the frequency.

- ON (1) This choice activates the automatic blanking function. The signal generator determines the blanking occurrences for optimum performance.
- OFF (0) This choice turns off the automatic blanking function, which also sets the blanking state to off.

#### Example

```
:OUTP:BLAN:AUTO 0
```

The preceding example disables RF output blanking.

```
*RST 1
```

**Key Entry** Output Blanking Off On Auto

### :BLANking:[STATe]

**Supported** All Models

```
[ :SOURce]:OUTPut:BLANking[:STATe] ON|OFF|1|0
[ :SOURce]:OUTPut:BLANking[:STATe]?
```

This command sets the state for RF Output blanking. Blanking occurs when the RF output is momentarily turned off as the sweep transitions from one frequency segment (band) to another, allowing the signal to settle. Blanking also occurs during the retrace, so the signal can settle before the next sweep. In CW mode, blanking occurs whenever you change the frequency.

- ON (1) This choice activates the blanking function. Blanking occurs on all frequency changes, including segment transitions and retrace
- OFF (0) This choice turns off the blanking function.

#### Example

```
:OUTP:BLAN:ON
```

The preceding example enables RF output blanking.

**Key Entry** Output Blanking Off On Auto

## **:MODulation[:STATe]**

**Supported** E8257D/E8663D with Option UNT, and E8267D

```
:OUTPut:MODulation[:STATe] ON|OFF|1|0  
:OUTPut:MODulation[:STATe]?
```

This command enables or disables the modulation of the RF output with the currently active modulation type(s). Most modulation types can be simultaneously enabled except FM with  $\Phi M$ .

An annunciator on the signal generator always displays to indicate whether modulation is on or off.

### **Example**

```
:OUTP:MOD 0
```

The preceding example disables RF modulation.

```
*RST 1  
Key Entry      Mod On/Off
```

## **:SETTled?**

**Supported** All Models

```
:OUTPut:SETTled?
```

This command returns the current state of the source settled line. A “1” return value indicates the source is settled. A “0” return value indicates the source is unsettled. This command does not wait for the source to settle before returning the state. This is different than the \*OPC? command which delays the command response until the operation is complete.

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

## **:SETTled:POLarity**

**Supported** All models

```
:OUTPut:SETTled:POLarity NORMal|INVerted  
:OUTPut:SETTled:POLarity?
```

This command sets the polarity of the source settled line (also referred to as: “lock and level”). When the active polarity is set to NORMal, the source settled line will be TTL low level whenever the source is settled. If the active polarity is set to INVerted, the source will be TTL high level whenever the source is settled. (This is the default condition.)

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

```
NORMal            This choice indicates the source is settled by setting this line low.  
INVerted          This choice indicates the source is settled by setting this line high. (This is the  
                  default condition.)  
*RST              NORMal
```

**Key Entry            Source Settled Polarity Norm Inv**

**:SETTled:RETRace**

Supported            All models

```
:OUTPut:SETTled:RETRace NORMal|INVerted
:OUTPut:SETTled:RETRace?
```

This command sets the source settled state (also referred to as: “lock and level”) when a step or list sweep is armed and waiting for an external trigger input. In the “NORMal” mode, the source settled output indicates “Settled”; in the “INVerted” mode, the source settled output indicates “Not Settled.” This command is coupled to the “:SETTled:POLarity” on page 66 command. If the “:SETTled:POLarity” on page 66 command is “INVerted” and the retrace mode is “INVerted”, the source settled output will be at a TTL low level when the step or list sweep is armed and waiting for an external trigger input. (This is the default condition.)

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

NORMal            This choice indicates the source is settled when waiting for an external trigger.

INVerted           This choice indicates the source is not settled when waiting for an external trigger. (This is the default condition.)

\*RST                NORMal

**Key Entry            Source Settled Retrace Norm Inv**

**:SETTled:RFOff**

Supported            All models

```
:OUTPut:SETTled:RFOff NORMal|INVerted
:OUTPut:SETTled:RFOff?
```

This command sets the source settled state when the RF is off. In normal operation, the source settled line (also referred to as: “lock and level”) indicates a “Settled” condition when the RF is off. In the “INVerted” mode, the source settled line indicates an “Not Settled” condition when the RF is off. This command is coupled to the “:SETTled:POLarity” on page 66 command. If the “:SETTled:POLarity” on page 66 command is “INVerted” and the RF off settled mode is “INVerted”, the source settled line will be TTL low level when the RF is off. (This is the default condition.)

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

NORMal            This choice indicates the source is settled when the RF is off.

INVerted           This choice indicates the source is unsettled when the RF is off. (This is the default condition.)

\*RST                NORMal

**Key Entry            RF Off Source Settled Polarity Norm Inv**

## **[ :STATe ]**

**Supported**            All Models

:OUTPut[ :STATe] ON|OFF|1|0

:OUTPut[ :STATe]?

This command enables or disables the RF output. Although you can configure and engage various modulations, no signal is available at the RF OUTPUT connector until this command is executed.

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

### **Example**

```
:OUTP ON
```

The preceding example turns on the signal generator's RF output.

```
*RST            0
```

**Key Entry**            **RF On/Off**

## Route Subsystem (:ROUTE:HARDware:DGENERator)

### :INPut:BPOLarity

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:INPut:BPOLarity POSitive|NEGative  
:ROUTE:HARDware:DGENERator:INPut:BPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the BURST GATE IN connector. This command performs the same function as “:IPOLarity:BGATE” on page 70.

#### Example

```
:ROUT:HARD:DGEN:INP:BPOL NEG
```

The preceding example sets up the signal generator to respond to a LOW level TTL signal at the BURST GATE IN connector.

**\*RST** POS

**Key Entry** Burst Gate In Polarity Neg Pos

### :INPut:CPOLarity

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:INPut:CPOLarity POSitive|NEGative  
:ROUTE:HARDware:DGENERator:INPut:CPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA CLOCK input connector. This command performs the same function as “:IPOLarity:CLOCK” on page 71.

#### Example

```
:ROUT:HARD:DGEN:INP:CPOL POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the DATA CLOCK input connector.

**\*RST** POS

**Key Entry** Data Clock Polarity Neg Pos

### :INPut:DPOLarity

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:INPut:DPOLarity POSitive|NEGative  
:ROUTE:HARDware:DGENERator:INPut:DPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA connector. This command performs the same function as “:IPOLarity:DATA” on page 71.

### Example

```
:ROUT:HARD:DGEN:INP:DPOL POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the DATA input connector.

**\*RST**                    POS

**Key Entry**            **Data Polarity Neg Pos**

### :INPut:SPOLarity

**Supported**            E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:INPut:SPOLarity POSitive|NEGative
```

```
:ROUTE:HARDware:DGENerator:INPut:SPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the SYMBOL SYNC input connector.

This command performs the same function as “[:IPOLarity:SSYNc](#)” on page 71.

### Example

```
:ROUT:HARD:DGEN:INP:SPOL POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the SYMBOL SYNC input connector.

**\*RST**                    POS

**Key Entry**            **Symbol Sync Polarity Neg Pos**

### :IPOLarity:BGATe

**Supported**            E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:IPOLarity:BGATe POSitive|NEGative
```

```
:ROUTE:HARDware:DGENerator:IPOLarity:BGATe?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL signal at the BURST GATE IN connector. This command performs the same function as “[:INPut:BPOLarity](#)” on page 69.

### Example

```
:ROUT:HARD:DGEN:IPOL:BGAT POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the rear panel BURST GATE IN connector.

**\*RST**                    POS

**Key Entry**            **Burst Gate In Polarity Neg Pos**

## **:IPOLarity:CLOCK**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:IPOLarity:CLOCK POSitive|NEGative  
:ROUTE:HARDware:DGENerator:IPOLarity:CLOCK?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA CLOCK connector.

This command performs the same function as “:INPut:CPOLarity” on page 69.

### **Example**

```
:ROUT:HARD:DGEN:IPOL:CLOC POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the DATA CLOCK input connector.

```
*RST POS
```

**Key Entry** Data Clock Polarity Neg Pos

## **:IPOLarity:DATA**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:IPOLarity:DATA POSitive|NEGative  
:ROUTE:HARDware:DGENerator:IPOLarity:DATA?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA connector. This command performs the same function as “:INPut:DPOLarity” on page 69

### **Example**

```
:ROUT:HARD:DGEN:IPOL:DATA POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the DATA input connector.

```
*RST POS
```

**Key Entry** Data Polarity Neg Pos

## **:IPOLarity:SSYNc**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:IPOLarity:SSYNc POSitive|NEGative  
:ROUTE:HARDware:DGENerator:IPOLarity:SSYNc?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the SYMBOL SYNC connector.

This command performs the same function as “:INPut:SPOLarity” on page 70.

### Example

```
:ROUT:HARD:DGEN:IPOL:SSYN POS
```

The preceding example sets up the signal generator to respond to a high level TTL signal at the SYMBOL SYNC input connector.

```
*RST          POS
```

**Key Entry**           **Symbol Sync Polarity Neg Pos**

### :OPOLarity:CLOCK

**Supported**           E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:OPOLarity:CLOCK POSitive|NEGative
```

```
:ROUTE:HARDware:DGENerator:OPOLarity:CLOCK?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the DATA CLK OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as [“:OUTPut:CPOLarity” on page 73](#).

### Example

```
:ROUT:HARD:DGEN:OPOL:CLOC POS
```

The preceding example sets up the signal generator to output a high level TTL signal at the DATA CLK OUT pin on the rear panel AUXILIARY I/O connector.

```
*RST          POS
```

**Key Entry**           **Data Clock Out Neg Pos**

### :OPOLarity:DATA

**Supported**           E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:OPOLarity:DATA POSitive|NEGative
```

```
:ROUTE:HARDware:DGENerator:OPOLarity:DATA?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the DATA OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as [“:OUTPut:DPOLarity” on page 74](#).

### Example

```
:ROUT:HARD:DGEN:OPOL:DATA NEG
```

The preceding example sets up the signal generator to output a low level TTL signal at the DATA OUT pin on the rear panel AUXILIARY I/O connector.

```
*RST          POS
```

**Key Entry**           **Data Out Polarity Neg Pos**



## **:OPOLarity:EVENT[1]|2|3|4**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:OPOLarity:EVENT[1]|2|3|4 POSitive|NEGative  
:ROUTE:HARDware:DGENERator:OPOLarity:EVENT[1]|2|3|4?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT 1 or EVENT 2 connector.

This command performs the same function as [“:OUTPut:EPOL\[1\]|2|3|4”](#) on page 75.

### **Example**

```
:ROUT:HARD:DGEN:OPOL:DATA NEG
```

The preceding example sets up the signal generator to output a low level TTL signal at the DATA OUT pin on the rear panel AUXILIARY I/O connector.

## **:OPOLarity:SSYNc**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:OPOLarity:SSYNc POSitive|NEGative  
:ROUTE:HARDware:DGENERator:OPOLarity:SSYNc?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level signal at the SYM SYNC OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as [“:OUTPut:SPOLarity”](#) on page 75.

### **Example**

```
:ROUT:HARD:DGEN:OPOL:SSYN POS
```

The preceding example sets up the signal generator to output a high level TTL signal at the SYM SYNC OUT pin on the rear panel AUXILIARY I/O connector.

**\*RST** POS

**Key Entry** **Symbol Sync Out Polarity Neg Pos**

## **:OUTPut:CPOLarity**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:OUTPut:CPOLarity POSitive|NEGative  
:ROUTE:HARDware:DGENERator:OUTPut:CPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the DATA CLK OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as [“:OPOLarity:CLOCK”](#) on page 72.

### **Example**

```
:ROUT:HARD:DGEN:OUTP:CPOL POS
```

The preceding example sets up the signal generator to output a high level TTL signal at the DATA CLOCK OUT pin on the rear panel AUXILIARY I/O connector.

**\*RST** POS

**Key Entry** Data Clock Polarity Neg Pos

### **:OUTPut:DCS[:STATe]**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:OUTPut:DCS[:STATe] ON|OFF|1|0
```

```
:ROUTE:HARDware:DGENerator:OUTPut:DCS[:STATe]?
```

This command is used to enable or disable the DATA OUT, DATA CLK OUT, and SYM SYNC OUT signals from the rear panel AUXILIARY I/O connector. Normally, these output signals should be enabled (On). However, disabling these outputs will decrease the spurs that are sometimes present when operating at high symbol rates.

#### **Example**

```
:ROUTE:HARD:DGEN:OUTP:DCS 1
```

The preceding example sets up or enables the DATA OUT, DATA CLK OUT, and SYM SYNC OUT output signals from the rear panel AUXILIARY I/O connector.

**\*RST** 1

**Key Entry** DATA/CLK/SYNC Rear Outputs Off On

### **:OUTPut:DPOLarity**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENerator:OUTPut:DPOLarity POSitive|NEGative
```

```
:ROUTE:HARDware:DGENerator:OUTPut:DPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the DATA OUT connector.

This command performs the same function as “[:OPOLarity:DATA](#)” on page 72.

#### **Example**

```
:ROUTE:HARD:DGEN:OUTP:DPOL POS
```

The preceding example sets up the signal generator to output a high level TTL signal at the DATA OUT connector.

**\*RST** POS

**Key Entry** Data Out Polarity Neg Pos

## **:OUTPut:EPOL[1]|2|3|4**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:OUTPut:EPOLarity[1]|2|3|4 POSitive|NEGative  
:ROUTE:HARDware:DGENERator:OUTPut:EPOLarity[1]|2|3|4?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT1 or EVENT 2 connector.

This command performs the same function as “:OPOLarity:EVENT[1]|2|3|4” on page 73.

### **Example**

```
:ROUT:HARD:DGEN:OUTP:EPOL1 POS
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT1 or EVENT 2 connector.

## **:OUTPut:SPOLarity**

**Supported** E8267D with Option 601 or 602

```
:ROUTE:HARDware:DGENERator:OUTPut:SPOLarity POSitive|NEGative  
:ROUTE:HARDware:DGENERator:OUTPut:SPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the SYMBOL SYNC connector.

### **Example**

```
:ROUT:HARD:DGEN:OUTP:SPOL POS
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT1 or EVENT 2 connector.

**\*RST** POS

**Key Entry** Symbol Sync Out Polarity Neg Pos

## Status Subsystem (:STATus)

### :OPERation:BASEband:CONDition

**Supported** E8267D with Option 601 or 602

:STATus:OPERation:BASEband:CONDition?

This query returns the decimal sum of the bits in the Baseband Operation Condition register. For example, if the baseband is busy (bit 0), the value 1 is returned.

The data in this register is continuously updated and reflects current signal generator conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

### :OPERation:BASEband:ENABle

**Supported** E8267D with Option 601 or 602

:STATus:OPERation:BASEband <val>  
:STATus:OPERation:BASEband:ENABle?

This command enables bits in the Baseband Operation Event Enable register. Bits enabled and set in this register will set bit 10 in the Standard Operation Condition register.

The variable <val> is the sum of the decimal values of the bits you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

#### Example

```
:STAT:OPER:BASE:ENAB 3
```

This command enables bit 0 (decimal 1, Baseband is Busy) and bit 1 (decimal 2, Baseband 1 Communicating) in the Baseband Operation Event Enable register.

**Range** 0 to 32767

### :OPERation:BASEband:NTRansition

**Supported** E8267D with Option 601 or 602

:STATus:OPERation:BASEband:NTRansition <val>  
:STATus:OPERation:BASEband:NTRansition?

This command enables bits in the Baseband Operation Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Baseband Operation Condition register will pass through and be read by the Baseband Operation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:OPER:BAS:NTR 3
```

This command enables bit 0 (decimal 1, Baseband 1 Busy) and bit 1 (decimal 2, Baseband 1 Communicating) in the Baseband Operation Negative Transition Filter register.

**Range** 0 to 32767

**:OPERation:BASEband:PTRansition**

**Supported** E8267D with Option 601 or 602

```
:STATus:OPERation:BASEband:PTRansition <val>
:STATus:OPERation:BASEband:PTRansition?
```

This command enables bits in the Baseband Operation Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Baseband Operation Condition register will pass through and be read by the Baseband Operation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:OPER:BAS:PTR 3
```

This command enables bit 0 (decimal 1, Baseband 1 Busy) and bit 1 (decimal 2, Baseband 1 Communicating) in the Baseband Operation Positive Transition Filter register.

**Range** 0 to 32767

**:OPERation:BASEband[:EVENT]**

**Supported** E8267D with Option 601 or 602

```
:STATus:OPERation:BASEband[:EVENT]?
```

---

**NOTE** This is a destructive read. The data in the Baseband Operation Event register is latched until it is queried. Once queried, the data is cleared.

---

This query returns the decimal sum of the bits in the Baseband Operation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## :OPERation:CONDition

**Supported** All Models

:STATus:OPERation:CONDition?

This query returns the decimal sum of the bits in the Standard Operation Condition register. This register monitors signal generator functions such as I/Q calibrating, sweeping, and measuring. For example, if a sweep is in progress (bit 3), a decimal 8 is returned with this query.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## :OPERation:ENABLE

**Supported** All Models

:STATus:OPERation:ENABle <val>

:STATus:OPERation:ENABle?

This command enables bits in the Standard Operation Event Enable register. Bits enabled and set in this register will set the Operation Status Summary bit (bit 7) in the Status Byte register. When bit 7 in the Status Byte register is set, you can read the Standard Operation Event register to determine the cause.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### Example

```
:STAT:OPER:ENAB 43
```

This command enables bit 0 (decimal 1, I/Q calibrating), bit 1 (decimal 2, Settling), bit 3 (decimal 8, Sweeping), and bit 5 (decimal 32, Waiting for Trigger) of the Standard Operation Event Enable register.

**Range** 0 to 32767

## :OPERation:NTRansition

**Supported** All Models

:STATus:OPERation:NTRansition <val>

:STATus:OPERation:NTRansition?

This command enables bits in the Standard Operation Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Standard Operation Condition register will pass through and be read by the Standard Operation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:OPER:NTR 3
```

This command enables bit 0 (decimal 1, I/Q Calibrating) and bit 1 (decimal 2, Settling) in the Standard Operation Negative Transition Filter register.

**Range**                    0 to 32767

**:OPERation:PTRansition**

**Supported**            All Models

```
:STATus:OPERation:PTRansition <val>
:STATus:OPERation:PTRansition?
```

This command enables bits in the Standard Operation Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Standard Operation Condition register will pass through and be read by the Standard Operation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:OPER:PTR 3
```

This command enables bit 0 (decimal 1, I/Q Calibrating) and bit 1 (decimal 2, Settling) in the Standard Operation Positive Transition Filter register.

**Range**                    0 to 32767

**:OPERation[:EVENT]**

**Supported**            All Models

---

**NOTE**    This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

```
:STATus:OPERation[:EVENT]?
```

This query returns the decimal sum of the bits in the Standard Operation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range**                    0 to 32767

## **:PRESet**

**Supported**            All Models

:STATus:PRESet

This command presets all positive and negative transition filters, enable registers, and error/event queue enable registers.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

## **:QUEStionable:CALibration:CONDition**

**Supported**            All Models

:STATus:QUEStionable:CALibration:CONDition?

This query returns the decimal sum of the bits in the Data Questionable Calibration Condition register. For example, if the DCFM or DCΦM zero calibration fails (bit 0), a value of 1 is returned.

The data in this register is continuously updated and reflects the current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range**                0 to 32767

## **:QUEStionable:CALibration:ENABle**

**Supported**            All Models

:STATus:QUEStionable:CALibration:ENABle <val>

:STATus:QUEStionable:CALibration:ENABle?

This command enables bits in the Data Questionable Calibration Event Enable register. Bits enabled and set in this register will set the Calibration Summary bit (bit 8) in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

:STAT:QUES:CAL:ENAB 1

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) in the Data Questionable Calibration Event Enable register.

**Range**                0 to 32767



## **:QUESTionable:CALibration:NTRansition**

**Supported**            All Models

```
:STATus:QUESTionable:CALibration:NTRansition <val>
:STATus:QUESTionable:CALibration:NTRansition?
```

This command enables bits in the Data Questionable Calibration Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Calibration Condition register will pass through and be read by the Data Questionable Calibration Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

```
:STAT:OPER:NTR 3
```

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) and bit 1 (decimal 2, I/Q Calibration Failure) in the Data Questionable Calibration Negative Transition Filter register.

**Range**                    0 to 32767

## **:QUESTionable:CALibration:PTRansition**

**Supported**            All Models

```
:STATus:QUESTionable:CALibration:PTRansition <val>
:STATus:QUESTionable:CALibration:PTRansition?
```

This command enables bits in the Data Questionable Calibration Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Calibration Condition register will pass through and be read by the Data Questionable Calibration Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

```
:STAT:OPER:PTR 3
```

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) and bit 1 (decimal 2, I/Q Calibration Failure) in the Data Questionable Calibration Positive Transition Filter register.

**Range**                    0 to 32767

## **:QUESTIONable:CALibration[:EVENT]**

**Supported** All Models

---

**NOTE** This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

:STATUS:QUESTIONable:CALibration[:EVENT]?

This command returns the decimal sum of the bits in the Data Questionable Calibration Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## **:QUESTIONable:CONDition**

**Supported** All Models

:STATUS:QUESTIONable:CONDition?

This query returns the decimal sum of the bits in the Data Questionable Condition register. For example, if the internal reference oscillator oven is cold (bit 4), a value of 16 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## **:QUESTIONable:ENABLE**

**Supported** All Models

:STATUS:QUESTIONable:ENABLE <val>  
:STATUS:QUESTIONable:ENABLE?

This command enables bits in the Data Questionable Event Enable register. Bits enabled and set in this register will set the Data Questionable Summary bit (bit 3) in the Status Byte register. When bit 3 in the Status Byte register is set, you can read the Data Questionable Event register to determine the cause.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

```
:STAT:QUES:ENAB 8
```

This command enables bit 3 (decimal 8, the Power Summary bit), in the Data Questionable Event Enable register.

**Range** 0 to 32767

**:QUESTionable:FREQuency:CONDition****Supported** All Models`:STATus:QUESTionable:FREQuency:CONDition?`

This query returns the decimal sum of the bits in the Data Questionable Frequency Condition register. For example, if the 1 GHz internal reference clock is unlocked (bit 2), a value of 4 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767**:QUESTionable:FREQuency:ENABLE****Supported** All Models`:STATus:QUESTionable:FREQuency:ENABLE <val>`  
`:STATus:QUESTionable:FREQuency:ENABLE?`

This command enables bits in the Data Questionable Frequency Event Enable register. Bits enabled and set in this register will set the Data Questionable Condition register bit 5.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**`:STAT:QUES:FREQ:ENAB 7`

This command enables bit 0 (decimal 1, Synthesizer Unlocked), bit 1 (decimal 2, 10 MHz Reference Unlocked), and bit 2 (decimal 4, 1 GHz reference Unlocked) in the Data Questionable Frequency Event Enable register.

**Range** 0 to 32767**:QUESTionable:FREQuency:NTRansition****Supported** All Models`:STATus:QUESTionable:FREQuency:NTRansition <val>`  
`:STATus:QUESTionable:FREQuency:NTRansition?`

This command enables bits in the Data Questionable Frequency Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Frequency Condition register will pass through and be read by the Data Questionable Frequency Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:QUES:FREQ:NTR 96
```

This command enables bit 5 (decimal 32, Sampler Loop Unlocked) and bit 6 (decimal 64, YO Loop Unlocked) in the Data Questionable Frequency Negative Transition Filter register.

**Range** 0 to 32767

**:QUESTIONable:FREQUENCY:PTRansition**

**Supported** All Models

```
:STATus:QUEStionable:FREQuency:PTRansition <val>  
:STATus:QUEStionable:FREQuency:PTRansition?
```

This command enables bits in the Data Questionable Frequency Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Frequency Condition register will pass through and be read by the Data Questionable Frequency Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:QUES:FREQ:PTR 8
```

This command enables bit 3 (decimal 8, Baseband 1 Unlocked) in the Data Questionable Frequency Positive Transition Filter register.

**Range** 0 to 32767

**:QUESTIONable:FREQUENCY[:EVENT]**

**Supported** All Models

---

**CAUTION** This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

```
:STATus:QUEStionable:FREQuency[:EVENT]?
```

This query returns the decimal sum of the bits in the Data Questionable Frequency Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## **:QUESTionable:MODulation:CONDition**

**Supported** All Models

```
:STATus:QUESTionable:MODulation:CONDition?
```

This command returns the decimal sum of the bits in the Data Questionable Modulation Condition register. For example, if the modulation is uncalibrated (bit 4), a value of 16 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

## **:QUESTionable:MODulation:ENABLE**

**Supported** All Models

```
:STATus:QUESTionable:MODulation:ENABLE <val>  
:STATus:QUESTionable:MODulation:ENABLE?
```

This command enables bits in the Data Questionable Modulation Event Enable register. Bits enabled and set in this register will set bit 7 in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

```
:STAT:QUES:MOD:ENAB 20
```

This command enables bit 2 (decimal 4, Modulation 1 Overmod) and bit 4 (decimal 16, Modulation Uncalibrated) in the Data Questionable Modulation Event Enable register.

**Range** 0 to 32767

## **:QUESTionable:MODulation:NTRansition**

**Supported** All Models

```
:STATus:QUESTionable:MODulation:NTRansition <val>  
:STATus:QUESTionable:MODulation:NTRansition?
```

This command enables bits in the Modulation Questionable Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Modulation Questionable Condition register will pass through and be read by the Modulation Questionable Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

### **Example**

```
:STAT:QUES:MOD:NTR 3
```

This command enables bit 0 (decimal 1, Modulation 1 Undermod) and bit 1 (decimal 2, Modulation 1 Overmod) in the Data Questionable Modulation Negative Transition Filter register.

**Range** 0 to 32767

### **:QUESTIONable:MODulation:PTRansition**

**Supported** All Models

```
:STATus:QUESTIONable:MODulation:PTRansition <val>  
:STATus:QUESTIONable:MODulation:PTRansition?
```

This command enables bits in the Data Questionable Modulation Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Modulation Condition register will pass through and be read by the Data Questionable Modulation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

#### **Example**

```
:STAT:QUES:MOD:PTR 3
```

This command enables bit 0 (decimal 1, Modulation 1 Undermod) and bit 1 (decimal 2, Modulation 1 Overmod) in the Data Questionable Modulation Positive Transition Filter register.

**Range** 0 to 32767

### **:QUESTIONable:MODulation[:EVENT]**

**Supported** All Models

---

**CAUTION** This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

```
:STATus:QUESTIONable:MODulation[:EVENT]?
```

This query returns the decimal sum of the bits in the Data Questionable Modulation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range** 0 to 32767

### **:QUESTIONable:NTRansition**

**Supported** All Models

```
:STATus:QUESTIONable:NTRansition <val>  
:STATus:QUESTIONable:NTRansition?
```

This command enables bits in the Data Questionable Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Condition register will pass through and be read by the Data Questionable Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:QUES:MOD:NTR 3072
```

This command enables bit 10 (decimal 1024, Baseband is busy) and bit 11 (decimal 2048, Sweep Calculating) in the Data Questionable Negative Transition Filter register.

**Range**                    0 to 32767

**:QUESTionable:POWer:CONDition**

**Supported**                All Models

```
:STATus:QUESTionable:POWer:CONDition?
```

This query returns the decimal sum of the bits in the Data Questionable Power Condition register. For example, if the RF output signal is unlevelled (bit 1), a value of 2 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Range**                    0 to 32767

**:QUESTionable:POWer:ENABle**

**Supported**                All Models

```
:STATus:QUESTionable:POWer:ENABle <val>
:STATus:QUESTionable:POWer:ENABle?
```

This command enables bits in the Data Questionable Power Event Enable register. Bits enabled and set in this register will set bit 3 in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

**Example**

```
:STAT:QUES:POW:ENAB 1
```

This command enables bit 0 (decimal 1, Reverse Power Protection Tripped) in the Data Questionable Power Event Enable register.

**Range**                    0 to 32767

## **:QUEStionable:POWer:NTRansition**

**Supported**            All Models

```
:STATus:QUEStionable:POWer:NTRansition <val>  
:STATus:QUEStionable:POWer:NTRansition?
```

This command enables bits in the Data Questionable Power Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Power Condition register will pass through and be read by the Data Questionable Power Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

### **Example**

```
:STAT:QUES:POW:NTR 1
```

This command enables bit 0 (Reverse Power Protection Tripped) in the Data Questionable Power Negative Transition Filter register.

**Range**                0 to 32767

## **:QUEStionable:POWer:PTRansition**

**Supported**            All Models

```
:STATus:QUEStionable:POWer:PTRansition <val>  
:STATus:QUEStionable:POWer:PTRansition?
```

This command enables bits in the Data Questionable Power Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Power Condition register will pass through and be read by the Data Questionable Power Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

### **Example**

```
:STAT:QUES:POW:PTR 1
```

This command enables bit 0 (decimal 1, Reverse Power Protection Tripped) in the Data Questionable Power Positive Transition Filter register.

**Range**                0 to 32767

## **:QUEStionable:POWer[:EVENT]**

**Supported**            All Models

---

**CAUTION**            This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

```
:STATus:QUEStionable:POWer[:EVENT]?
```



This query returns the decimal sum of the bits in the Data Questionable Power Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

**Range** 0 to 32767

### **:QUESTIONable:PTRansition**

**Supported** All Models

```
:STATus:QUEStionable:PTRansition <val>
:STATus:QUEStionable:PTRansition?
```

This command enables bits in the Data Questionable Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Condition register will pass through and be read by the Data Questionable Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

#### **Example**

```
:STAT:QUES:PTR 8
```

This command enables bit 3 (decimal 8, Power Summary) in the Data Questionable Positive Transition Filter register.

**Range** 0 to 32767

### **:QUESTIONable[:EVENT]**

**Supported** All Models

---

**CAUTION** This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

---

```
:STATus:QUEStionable[:EVENT]?
```

This query returns the decimal sum of the bits in the Standard Operation Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

**Range** 0 to 32767

## System Subsystem (:SYSTem)

### :ALternate

**Supported** All with Option 007

```
:SYSTem:ALternate <reg_num>  
:SYSTem:ALternate? [MAXimum|MINimum]
```

This command sets up the signal generator to use a sweep state stored in a state register to alternate with the current sweep. The alternate sweep state must be stored in state registers 1 through 9 in sequence 0. Alternate sweep must be selected and both sweeps must be ramp sweeps.

#### Example

```
:SYST:ALT 3
```

The preceding example alternates the current sweep with the sweep settings saved in state register number three.

**Key Entry** Alternate Sweep Seq 0, Register 1–9

### :ALternate:STAtE

**Supported** All with Option 007

```
:SYSTem:ALternate:STATe ON|OFF|1|0  
:SYSTem:STATe?
```

This command enables or disables the alternate sweep state for the signal generator. With alternate state on, the signal generator uses the current sweep setup and alternates with a sweep saved in one of the state registers. Both sweeps must be ramp sweeps.

#### Example

```
:SYST:ALT:STAT OFF
```

The preceding example disables the alternate sweep mode.

**Key Entry** Alternate Sweep Off On

### :CAPability

**Supported** All Models

```
:SYSTem:CAPability?
```

This query returns the signal generator's capabilities and outputs the appropriate specifiers:

```
(RFSOURCE WITH( (AM|FM|PULM|PM|LFO)&(FSSWEEP|FLIST)&(PSSWEEP|PLIST)  
&TRIGGER&REFERENCE))
```

This is a list of the SCPI-defined basic functionality of the signal generator and the additional capabilities it has in parallel (a&b) and singularly (a|b).

## :DATE

**Supported** All Models

```
:SYSTem:DATE <year>, <month>, <day>
:SYSTem:DATE?
```

This command sets the date as shown in the lower right area of the signal generator display.

<year> This variable requires a four digit integer.

The query returns the date in the following format: <+year>, <+month>, <+day>

### Example

```
:SYST:DATE 2004,12,15
```

The preceding example sets the date.

**Range** <month>: 1-12 <day>: 1-31

**Key Entry** Time/Date

## :ERRor[:NEXT]

**Supported** All Models

```
:SYSTem:ERRor[:NEXT]?
```

This query returns the most recent error message from the signal generator error queue. If there are no error messages, the query returns the following output:

```
+0, "No error"
```

When there is more than one error message, the query will need to be sent for each message.

The error messages are erased after being queried.

**Key Entry** Error Info View Next Error Message

## :ERRor:SCPI[:SYNTax]

**Supported** All Models

```
:SYSTem:ERRor:SCPI[:SYNTax] ON|OFF|1|0
:SYSTem:ERRor:SCPI[:SYNTax]?
```

This command allows you to turn on verbose error messages that point out where the SCPI parser generated an error. Use the `ERRor[:NEXT]` command to read any reported errors.

---

**CAUTION** When this command is enabled, the instrument's speed is slowed. Where speed is important, disable this feature.

---

### Example

```
:SYST:ERR:SCPI ON
```

The preceding example enables the SCPI command error report function.

```
*RST 0
```

## :FILEsystem:SAFEmode

Supported All Models

```
:SYSTem:FILEsystem:SAFEmode ON|OFF|1|0  
:SYSTem:FILEsystem:SAFEmode?
```

This command selects the safe mode for file handling. When safe mode is set to OFF, volatile waveform files can be edited and saved while the signal generator plays the file without signal interruption. However, it is possible with complex waveforms, for corruption of memory to occur which will be reported as an error on the front panel display and require a reboot of the signal generator to resolve.

### Example

```
:SYST:FILE:SAVE ON
```

The preceding example enables the safe mode setting and waveform files cannot be edited without signal disruption while the signal generator plays them.

\*RST On

## :HELP:MODE

Supported All Models

```
:SYSTem:HELP:MODE SINGLE|CONTInuous  
:SYSTem:HELP:MODE?
```

This command sets the help function mode of the signal generator.

SINGLE Help is provided only for the next key that you press.

CONTInuous Help is provided for each key you press. In addition, the function of the key is executed.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:SYST:HELP:MODE CONT
```

The preceding example enables the Help system continuous mode.

**Key Entry Help Mode Single Cont**

## :IDN

Supported All Models

```
:SYSTem:IDN "string"
```

This command modifies the identification string that the \*IDN? query returns. Sending an empty string returns the query output of \*IDN? to its factory-shipped setting. The maximum string length is 72 characters.

Modification of the \*IDN? query output enables the signal generator to identify itself as another signal generator when used as a replacement.

The display diagnostic information, shown by pressing the **Diagnostic Info** softkey, is not affected by this command.

## :LANGuage

**Supported** All Models

```
:SYSTem:LANGuage "SCPI"|"COMP"|"8340"|"8360"|"83712"|"83732"|"83752"|"8757"  
|"8662"|"8663"  
:SYSTem:LANGuage?
```

This command sets the remote language for the signal generator.

SCPI	This choice provides compatibility for SCPI commands.
COMP	This choice provides compatibility for a system, comprising a PSG signal generator and a 8757D scalar network analyzer. It is supported only through a GPIB interface.
8340	This choice provides compatibility for 8340B and 8341B microwave sources, which are supported by using the GPIB interface.
8360	This choice provides compatibility for 8360 series swept signal generators, which are supported only through a GPIB interface.
83712	This choice provides compatibility for 83711B and 83712B synthesized CW generators, which are supported only through a GPIB interface.
83732	This choice provides compatibility for 83731B and 83732B synthesized signal generators, which are supported only through a GPIB interface.
83752	This choice provides compatibility for 83751B and 83752B synthesized sweepers, which are supported only through a GPIB interface.
8757	This choice provides compatibility for a system, comprising a PSG signal generator and a 8757D scalar network analyzer. It is supported only through a GPIB interface.
8662	This choice provides compatibility for the Agilent 8662A Synthesized Signal Generator. The 8662A is controlled only through a GPIB interface.
8663	This choice provides compatibility for the Agilent 8663A Synthesized Signal Generator. The 8663A is controlled only through a GPIB interface.

The setting enabled by this command is not affected by a power-on, preset, or \*RST command.

For more information on supported SCPI commands and programming codes, refer to [Chapter 7, "SCPI Command Compatibility,"](#) on page 347.

### Example

```
:SYST:LANG "8757"
```

The preceding example enables the 8757 Network Analyzer language as the language used to control the signal generator.

<b>Key Entry</b>	<b>SCPI</b>	<b>8757D System</b>	<b>8340B, 8341B</b>	<b>83731B, 83732B</b>
	<b>8360 Series</b>	<b>8662A, 8663A</b>	<b>83711B, 83712B</b>	<b>83751B, 83752B</b>

### **:OEMHead:FREQuency:STARt**

**Supported** All Models

```
:SYSTem:OEMHead:FREQuency:STARt <val>  
:SYSTem:OEMHead:FREQuency:STARt?
```

This command sets the start frequency or minimum band frequency for an external source module. The pre-defined start or minimum band frequency for the selected WR (waveguide rectangular) is overwritten with this command. For more information on pre-defined frequency bands, refer to “:OEMHead:FREQuency:BAND WR15|WR12|WR10|WR8|WR6|WR5|WR3” on page 95.

#### **Example**

```
:SYST:OEMH:FREQ:STAR 90GHZ
```

The preceding example sets the start frequency for the OEM module to 90 GHz.

```
*RST 5.00000000000000E+10
```

**Key Entry** **Min Band Freq**

### **:OEMHead:FREQuency:STOP**

**Supported** All Models

```
:SYSTem:OEMHead:FREQuency:STOP <val>  
:SYSTem:OEMHead:FREQuency:STOP?
```

This command sets the stop frequency or maximum band frequency for an external source module. The pre-defined stop or maximum band frequency for the selected WR (waveguide rectangular) is overwritten with this command. For more information on pre-defined frequency bands, refer to “:OEMHead:FREQuency:BAND WR15|WR12|WR10|WR8|WR6|WR5|WR3” on page 95.

#### **Example**

```
:SYST:OEMH:FREQ:STOP 70GHZ
```

The preceding example sets the stop frequency for the OEM module to 70 GHz.

```
*RST 7.00000000000000E+10
```

**Key Entry** **Max Band Freq**

### **:OEMHead:SElect**

**Supported** All Models

```
:SYSTem:OEMHead:SElect ON|OFF|NONE|REAR|FRONT  
:SYSTem:OEMHead:SElect?
```

This command selects an external millimeter-wave source module. The ON, REAR, and FRONT

parameters select the OEM source module while the OFF and NONE parameters deselect the OEM source module. The MMOD and MULT annunciators, in the signal generator's frequency display will appear when a OEM millimeter-wave source module is selected.

**Example**

```
:SYST:OEMH:SEL ON
```

The preceding example turns on the OEM source module.

```
*RST Off
```

**Key Entry OEM Source Module Off On**

**:OEMHead:FREQuency:BAND WR15 | WR12 | WR10 | WR8 | WR6 | WR5 | WR3**

**Supported All Models**

```
:SYSTem:OEMHead:FREQuency:BAND WR3
:SYSTem:OEMHead:FREQuency:BAND?
```

This command allows you to select a pre-defined waveguide rectangular (WR) frequency band. The WR selection is determined by the external millimeter-wave source module frequency range. Selection of a WR frequency band sets the minimum and maximum frequency bands, for the external mm-wave source module, to pre-defined values shown in the table below. These pre-defined frequency bands are common to commercially available mixers and multipliers. Different start, stop, and multiplier values can be selected from the menu displayed under the **OEM Source Module Config** softkey.

**Table 2-2**

Waveguide Band	PSG Start Frequency	PSG Stop Frequency	Multiplier
WR15 50–75GHz	12.5000000000 GHz	18.7500000000 GHz	4.000 x
WR12 60–90GHz	10.0000000000 GHz	15.0000000000 GHz	6.000 x
WR10 75–110GHz	12.5000000000 GHz	18.4000000000 GHz	6.000 x
WR8 90–140GHz	11.2200000000 GHz	17.5000000000 GHz	8.000 x
WR6 110–170GHz	9.1000000000 GHz	14.2000000000 GHz	12.000 x
WR5 140–220GHz	11.6000000000 GHz	18.4000000000 GHz	12.000 x
WR3 220–325GHz	12.2000000000 GHz	18.1000000000 GHz	18.000 x

**Example**

```
:SYST:OEMH:FREQ:BAND WR12
```

The preceding example selects the 60–90 GHz WR frequency band.

```
*RST WR15
```

**Key Entry WR15 50–75GHz**

## **:OEMHead:FREQuency:MULTiplier**

**Supported** All Models

```
:SYSTem:OEMHead:FREQuency:MULTiplier <val>  
:SYSTem:OEMHead:FREQuency:MULTiplier?
```

This command selects a multiplier for an external millimeter-wave source module.

The multiplier factor allows the signal generator's frequency display to show the source module's frequency. The selection is valid only when the OEM source module is selected. The signal generator's actual RF frequency is not changed by the multiplier. For example, if the signal generator's RF frequency is 20 GHz and a 4.000 x multiplier is selected, the signal generator will display 80 GHz.

The displayed frequency on the signal generator is affected if the frequency reference and frequency offset settings. The relationship is described as follows: Displayed Frequency = (Actual Freq - Freq Reference) \* Frequency Multiplier + Freq Offset. Refer to the [:FREQuency:OFFSet](#) and [:FREQuency:REFerence](#) command descriptions for more information.

### **Example**

```
:SYST:OEMH:FREQ:MULT 4
```

The preceding example selects a 4x multiplier so that the signal generator display shows the frequency at the output of the mm-wave source module.

```
*RST 4.00000000E+000
```

**Key Entry** **Freq Multiplier**

## **:PON:TYPE**

**Supported** All Models

```
:SYSTem:PON:TYPE PRESet | LAST  
:SYSTem:PON:TYPE?
```

This command sets the defined conditions for the signal generator at power on.

**PRESet** This choice sets the conditions to factory- or user-defined as determined by the choice for the preset type. Refer to the [:PRESet:TYPE](#) command for selecting the type of preset.

**LAST** This choice retains the settings at the time the signal generator was last powered down.

The selection is not affected by a signal generator power-on, preset, or the \*RST command.

---

**NOTE** When LAST is selected, no signal generator interaction can occur for at least 3 seconds prior to cycling the power for the current settings to be saved.

---



## Example

```
:SYST:PON:TYPE PRES
```

The preceding example sets the preset state for the signal generator to factory settings.

**Key Entry**            **Power On Last Preset**

## :PRESet

**Supported**            All Models

```
SYSTem:PRESet
```

This command returns the signal generator to a set of defined conditions. It is equivalent to pressing the front panel **Preset** hardkey.

The defined conditions are either factory- or user-defined. Refer to the [:PRESet:TYPE](#) command for selecting the type of defined conditions.

**Key Entry**            **Preset**

## :PRESet:ALL

**Supported**            All Models

```
:SYSTem:PRESet:ALL
```

This command sets all states of the signal generator back to their factory default settings, including states that are not normally affected by a signal generator power-on, preset, or \*RST command.

## :PRESet:LANGuage

**Supported**            All Models

```
:SYSTem:PRESet:LANGuage "SCPI"|"8340"|"8360"|"83712"|"83732"|"83752" |  
"8757"|"8662"|"8663" |  
:SYSTem:PRESet:LANGuage?
```

This command sets the remote language that is available when the signal generator is preset.

SCPI	This choice provides compatibility for SCPI commands.
8340	This choice provides compatibility for 8340B and 8341B microwave sources, which are supported by using the GPIB interface.
8360	This choice provides compatibility for 8360 series swept signal generators, which are supported only through a GPIB interface.
83712	This choice provides compatibility for 83711B and 83712B synthesized CW generators, which are supported only through a GPIB interface.
83732	This choice provides compatibility for 83731B and 83732B synthesized signal generators, which are supported only through a GPIB interface.
83752	This choice provides compatibility for 83751B and 83752B synthesized sweepers,

which are supported only through a GPIB interface.

- 8757 This choice provides compatibility for a system, comprising a PSG signal generator and a 8757D scalar network analyzer. It is supported only through a GPIB interface.
- 8662 This choice provides compatibility for 8662A series synthesized waveform generators, which are supported only through a GPIB interface.
- 8663 This choice provides compatibility for 8663A series synthesized waveform generators, which are supported only through a GPIB interface.

**Example**

```
:SYST:PRES:LANG "8340"
```

The preceding example selects 8340 signal generator language as the language used by the signal generator following an instrument preset.

**\*RST "SCPI"**

<b>Key Entry</b>	<b>SCPI</b>	<b>8757D System</b>	<b>8340B, 8341B</b>	<b>83731B, 83732B</b>
	<b>8360 Series</b>	<b>8662A, 8663A</b>	<b>83711B, 83712B</b>	<b>83751B, 83752B</b>

**:PRESet:PERSistent**

**Supported** All Models

```
:SYSTem:PRESet:PERSistent
```

This command sets the states that are not affected by a signal generator power-on, preset, or \*RST command to their factory default settings.

**Key Entry** **Restore Sys Defaults**

**:PRESet:PN9**

**Supported** All Models

```
:SYSTem:PRESet:PN9 NORMal|QUICK  
:SYSTem:PRESet:PN9?
```

This command sets the preset length of the PN9 sequence for personalities that require software PRBS generation.

**NORMal** This choice produces a maximal length PN9 sequence.

**QUICK** This choice produces a truncated (216 bits) PN9 sequence.

**Example**

```
:SYST:PRES:PN9 NORMAL
```

The preceding example selects a maximum length PN9 sequence.

**\*RST** **NORM**

**Key Entry** **PN9 Mode Preset Normal Quick**

## **:PRESet:TYPE**

**Supported**            All Models

```
:SYSTem:PRESet:TYPE NORMal|USER
:SYSTem:PRESet:TYPE?
```

This command toggles the preset state between factory- and user-defined conditions. Refer to the [:PRESet\[:USER\]:SAVE](#) command for saving the USER choice preset settings. The setting enabled by the command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:SYST:PRESet:TYPE USER
```

The preceding example selects a user defined conditions for the signal generator preset state.

**Key Entry**            **Preset Normal User**

## **:PRESet[:USER]:SAVE**

**Supported**            All Models

```
:SYSTem:PRESet[:USER]:SAVE
```

This command saves your user-defined preset conditions to a state file.

Only one user-defined preset file can be saved. Subsequent saved user-defined preset files will overwrite the previously saved file.

**Key Entry**            **Save User Preset**

## **:SECurity:DISPlay**

**Supported**            All Models

```
:SYSTem:SECurity:DISPlay ON|OFF|1|0
:SYSTem:SECurity:DISPlay?
```

This command enables or disables the secure display mode.

**On(1)**                This selection turns the signal generator display back on, showing the current settings. Cycling the signal generator power also restores the display, however the current settings may change depending on the power-on configuration choice. Refer to the [:PON:TYPE](#) command for information on the power-on choices available.

**OFF(0)**                This selection blanks the signal generator's display, hiding the settings and disabling the front panel keys. While in this mode, the display shows  
\*\*\* SECURE DISPLAY ACTIVATED \*\*\*.

For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

**Example**

```
:SYST:SEC:DISP OFF
```

The preceding example enables the secure display mode.

```
*RST 1
```

**Key Entry            Activate Security Display**

**:SECurity:ERASeall**

**Supported            All Models**

```
:SYSTem:SECurity:ERASall
```

This command removes all user files, flatness correction files, and baseband generator files. In addition, all table editor files are returned to their original factory values.

This command differs from the :DELeTe:ALL command, which does not reset table editors to factory values. For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

**Key Entry            Erase All**

**:SECurity:LEVel**

**Supported            All Models**

```
:SYSTem:SECurity:LEVel NONE|ERASe|OVERwrite|SANitize  
:SYSTem:SECurity:LEVel?
```

This command selects the security level operation for the signal generator.

- NONE            This selection causes the signal generator to reset to factory default settings.
- ERASe           This selection removes all user files, table editor files, flatness correction files, and baseband generator files.
- OVERwrite       This selection removes all user files, table editor files, flatness correction files, and baseband generator files. The memory is then overwritten with random data.
  - SRAM            All addressable locations will be overwritten with random characters.
  - Hard Disk       All addressable locations will be overwritten with random characters.
  - Flash Memory   The flash blocks will be erased.
- SANitize        This selection removes all user files, table editor files, flatness correction files, and baseband generator files using the same techniques as the OVERwrite selection for SRAM and flash memory. For the hard disk, the signal generator overwrites all addressable locations with a single character, its complement, and then with a random character.

Once you select the security level, you *must* execute the command from `:SECurity:LEVel:STATe` to arm the security level.

---

**NOTE** Once you select a security level and arm it, you cannot change the level.

---

For other cleaning and security operation descriptions, see also the `:SECurity:ERASeall`, the `:SECurity:OVERwrite`, and the `:SECurity:SANitize` commands. For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

**Example**

```
:SYST:SEC:LEV NONE
```

The preceding example sets the secure mode so it resets the signal generator to factory settings after completing the security operation.

**Key Entry**            **None Erase Overwrite Sanitize**

**:SECurity:LEVel:STATe**

**Supported**            All Models

---

**CAUTION** Ensure that you select the security level prior to executing this command with the ON (1) selection. Once you enable the state, you cannot reduce the security level.

---

```
:SYSTem:SECurity:LEVel:STATe ON|OFF|1|0
:SYSTem:SECurity:LEVel:STATe?
```

This command arms and executes the current security level parameter.

- On (1)                    This selection arms and prevents any changes to the current security level. Refer to `:SECurity:LEVel` command for setting the security level.
- OFF (0)                 This selection performs the actions required for the current security level setting. Cycling the signal generator power also performs the same function.

For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

**Example**

```
:SYST:SEC:LEV:STAT ON
```

The preceding example arms the secure mode selected with the `SYSTem:SECurity:LEVel` command.

**Key Entry**            **Enter Secure Mode**

## **:SECurity:OVERwrite**

**Supported** All Models

:SYSTem:SECurity:OVERwrite

This command removes all user files, table editor files values, flatness correction files, and baseband generator files. The memory is then overwritten with random data as described below. For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

SRAM All addressable locations will be overwritten with random characters.

HARD DISK All addressable locations will be overwritten with random characters.

FLASH MEMORY The flash blocks will be erased.

**Key Entry** Erase and Overwrite All

## **:SECurity:SANitize**

**Supported** All Models

:SYSTem:SECurity:SANitize

This command removes all user files, table editor files values, flatness correction files, and baseband generator files. The memory is then overwritten with a sequence of data as described below. For more information about security functions, refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide*.

SRAM All addressable locations will be overwritten with random characters.

HARD DISK All addressable locations will be overwritten with a single character and then a random character.

FLASH MEMORY The flash blocks will be erased.

**Key Entry** Erase and Sanitize All

## **:SSAVer:DELay**

**Supported** All Models

:SYSTem:SSAVer:DELay <val>

:SYSTem:SSAVer:DELay?

This command sets the amount of time before the display light or display light and text is switched off. The time delay represents the time during which there is no signal generator front panel input. The variable <val> is a positive integer number, in hours. The setting enabled by this command is not affected by power-on, preset, or \*RST. See also the [:SSAVer:MODE](#) command for selecting the screen saver mode.

### **Example**

:SYST:SSAV:DEL 2

The preceding example sets two hours delay time for the screen saver mode.

**Range** 1–12  
**Key Entry** **Screen Saver Delay**

### **:SSAVer:MODE**

**Supported** All Models

```
:SYSTem:SSAVer:MODE LIGHT|TEXT
:SYSTem:SSAVer:MODE?
```

This command toggles the screen saver mode between light only or light and text.

**LIGHT** Enables only the light to turn off during the screen saver operation while leaving the text visible on the darkened screen.

**TEXT** Enables both the display light and text to turn off during screen saver operation.

The setting is not affected by a signal generator power-on, preset, or \*RST command.

#### **Example**

```
:SYST:SSAV:MODE TEXT
```

The preceding example sets the screen saver mode.

**Key Entry** **Screen Saver Mode**

### **:SSAVer:STATE**

**Supported** All Models

```
:SYSTem:SSAVer:STATE ON|OFF|1|0
:SYSTem:SSAVer:STATE?
```

This command enables or disables the display screen saver. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

#### **Example**

```
:SYST:SSAV:STAT 1
```

The preceding example enables the screen saver mode.

**Key Entry** **Screen Saver Off On**

### **:TIME**

**Supported** All Models

```
:SYSTem:TIME <hour>,<minute>,<second>
:SYSTem:TIME?
```

This command sets the time displayed in the lower right area of the signal generator's display.

**Range** **<hour>: 0–23 <minute>: 0–59 <second>: 0–59**

### Example

```
:SYST:TIME 9,30,45
```

The preceding example sets the signal generator time to 09:30:45.

<b>Key Entry</b>	<b>Time/Date</b>
------------------	------------------

### :VERSion

<b>Supported</b>	All Models
------------------	------------

```
:SYSTem:VERSion?
```

This command returns the SCPI version number with which the signal generator complies.



## Trigger Subsystem

### :ABORt

**Supported**            All Models

:ABORt

This command causes the List or Step sweep in progress to abort. If `INIT:CONT[:ALL]` is set to ON, the sweep will immediately re-initiate. The pending operation flag affecting `*OPC`, `*OPC?`, and `*WAI` will undergo a transition once the sweep has been reset.

### :INITiate:CONTInuous[:ALL]

**Supported**            All Models

:INITiate:CONTInuous[:ALL] ON|OFF|1|0

:INITiate:CONTInuous[:ALL]?

This command selects either a continuous or single List or Step sweep. Execution of this command does not affect a sweep in progress.

ON (1)                    Selects continuous sweep where, after the completion of the previous sweep, the sweep restarts automatically, or waits for a trigger.

OFF (0)                   This choice selects a single sweep. Refer to the [:INITiate:IMMediate\[:ALL\]](#) command for single sweep triggering information.

#### Example

```
:INIT:CONT ON
```

The preceding example enables the continuous mode for the sweep type.

```
*RST                    0
```

**Key Entry**            **Sweep Repeat Single Cont**

### :INITiate[:IMMediate][:ALL]

**Supported**            All Models

:INITiate[:IMMediate][:ALL]

This command either sets or sets and starts a single List or Step sweep, depending on the trigger type. The command performs the following:

- arms a single sweep when `BUS`, `EXTernal`, or `KEY` is the trigger source selection
- arms and starts a single sweep when `IMMediate` is the trigger source selection

This command is ignored if a sweep is in progress. For setting a continuous or single sweep, refer to the [:INITiate:CONTInuous\[:ALL\]](#) command. The trigger source is selected by the [:TRIGger\[:SEQuence\]:SOURce](#) command.

In some atypical cases, the :INIT command could be ignored if it immediately follows an \*OPC? command. If the :INIT command is ignored, then use a 10ms sleep function before sending the command.

**Key Entry**            **Single Sweep**

### **:TRIGger:OUTPut:POLarity**

**Supported**            All Models

```
:TRIGger:OUTPut:POLarity POSitive|NEGative  
:TRIGger:OUTPut:POLarity?
```

Sets the TTL signal level present at the TRIGGER OUT connector to either high (5 vdc) or low (0 vdc). The trigger out is asserted after the frequency and/or power is set while the sweep is waiting for its step trigger. In addition, the swept-sine sends a pulse to the TRIGGER OUT at the beginning of each sweep.

#### **Example**

```
:TRIG:OUTP:POL NEG
```

The preceding example enables the continuous mode as the sweep type.

```
*RST                    POS
```

**Key Entry**            **Trigger Out Polarity Neg Pos**

### **:TRIGger[:SEquence]:SLOPe**

**Supported**            All Models

```
:TRIGger[:SEquence]:SLOPe POSitive|NEGative  
:TRIGger[:SEquence]:SLOPe?
```

This command sets the polarity of the ramp or sawtooth waveform slope present at the TRIGGER IN connector that will trigger a List or Step sweep.

#### **Example**

```
:TRIG:SLOP POS
```

The preceding example sets a positive ramp slope.

```
*RST                    POS
```

**Key Entry**            **Trigger In Polarity Neg Pos**

### **:TRIGger[:SEquence]:SOURce**

**Supported**            All Models

```
:TRIGger[:SEquence]:SOURce BUS|IMMediate|EXTernal|KEY  
:TRIGger[:SEquence]:SOURce?
```

This command sets the sweep trigger source for a List or Step sweep.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN and RS-232 triggering using the *TRG command.
IMMEDIATE	This choice enables immediate triggering of the sweep event.
EXTERNAL	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables front panel triggering by pressing the <b>Trigger</b> hardkey.

The wait for the BUS, EXTERNAL, or KEY trigger can be bypassed by sending the :TRIGGER[:SEQUENCE][:IMMEDIATE] command.

### Example

```
:TRIG:SOUR BUS
```

The preceding example sets the sweep trigger source to BUS.

```
*RST          IMM
Key Entry    Bus    Free Run    Ext    Trigger Key
```

### :TRIGGER[:SEQUENCE][:IMMEDIATE]

**Supported**      All Models

```
:TRIGGER[:SEQUENCE][:IMMEDIATE]
```

This event command causes an armed List or Step sweep to immediately start without the selected trigger occurring.

In some atypical cases, the :TRIG command could be ignored if it immediately follows an \*OPC? command. If the :TRIG command is ignored, then use a 10ms sleep function before sending the command.

## Unit Subsystem (:UNIT)

### :POWer

**Supported**            All Models

```
:UNIT:POWer DBM|DBUV|DBUVEMF|V|VEMF|DB  
:UNIT:POWer?
```

This command terminates an amplitude value in the selected unit of measure.

If the amplitude reference state is set to on, the query returns units expressed in dB. Setting any other unit will cause a setting conflict error stating that the amplitude reference state must be set to off. Refer to the [:REfERENCE:STATE](#) command for more information.

All power values in this chapter are shown with DBM as the unit of measure. If a different unit of measure is selected, replace DBM with the newly selected unit whenever it is indicated for the value.

#### Example

```
:UNIT:POW DBM
```

The preceding example selects dBm as the unit of amplitude measurement.

**\*RST**                    DBM

Key Entry	dBm	dBuV	dBuVemf	mV	uV	mVemf	uVemf
-----------	-----	------	---------	----	----	-------	-------

---

## 3 Basic Function Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to signal generator operations common to all PSG models:

- “Correction Subsystem ([:SOURCE]:CORRection)” on page 110
- “Frequency Subsystem ([:SOURCE])” on page 113
- “List/Sweep Subsystem ([:SOURCE])” on page 128
- “Marker Subsystem–Option 007 ([:SOURCE])” on page 139
- “Power Subsystem ([:SOURCE]:POWer)” on page 142
- “Trigger Sweep Subsystem ([:SOURCE])” on page 155

## Correction Subsystem ([:SOURce]:CORRection)

### :FLATness:LOAD

**Supported** All Models

```
[:SOURce]:CORRection:FLATness:LOAD "<file_name>"
```

This command loads a user flatness correction file designated by the file name "<file\_name>" variable. The file will be loaded from the signal generator's USERFLAT directory. The directory path does not need to be specified in the command. Refer to the *Agilent Signal Generators Programming Guide* for more information on flatness corrections.

For information on file name syntax, refer to "File Name Variables" on page 10.

#### Example

```
:CORR:FLAT:LOAD "Flatness_Data"
```

The preceding example loads a user flatness file named "Flatness\_Data" from the signal generator's user flatness directory.

**Key Entry** Load From Selected File

### :FLATness:PAIR

**Supported** All Models

```
[:SOURce]:CORRection:FLATness:PAIR <freq>,<corr>
```

This command adds or edits a frequency and amplitude correction pair. The maximum number of pairs or points that can be entered is 3201. *Refer to the Agilent Signal Generators Programming Guide* for more information on flatness corrections. The <corr> variable is the power correction in dB.

Power and frequency ranges for different signal generator models and options are listed on page 153.

#### Example

```
:CORR:FLAT:PAIR 10MHz, .1
```

The preceding example enters a frequency of 10 megahertz and a power of 0.1dB into the user flatness table.

**\*RST**                    **Option 520:** +2.0000000000000E+10  
Option 532: +3.2000000000000E+10  
Option 540: +4.0000000000000E+10  
Option 544: +4.4000000000000E+10  
Option 550: +5.0000000000000E+10  
Option 567: +6.7000000000000E+10

**Range**                    **Option 520:** 250kHz–20GHZ  
Option 532: 250kHz–32GHZ  
Option 540: 250kHz–40GHZ  
Option 544: 250kHz–44GHZ  
Option 550: 250kHz–50GHZ  
Option 567: 250kHz–70GHZ<sup>a</sup>

a.67–70 GHz performance not specified

**Key Entry**            **Configure Cal Array**

**:FLATness:POINts**

**Supported**            All Models

`[:SOURCE]:CORREction:FLATness:POINts?`

This query returns the number of points in the user flatness correction file.

**:FLATness:PRESet**

**Supported**            All Models

---

**CAUTION**        Once this command is executed, correction data is overwritten; If needed, save the current correction data (see [“:FLATness:STORE”](#) on page 111).

---

`[:SOURCE]:CORREction:FLATness:PRESet`

This command presets the user flatness correction to a factory-defined setting that consists of one frequency point and one amplitude point with no corrections.

**Key Entry**            **Preset List**

**:FLATness:STORE**

**Supported**            All Models

`[:SOURCE]:CORREction:FLATness:STORE "<file_name>"`

This command stores the current user flatness correction data to a file named by the "`<file_name>`" variable. All user flatness files are stored in the signal generator's USERFLAT directory. The directory path does not need to be specified in the command.

For information on file name syntax, refer to [“File Name Variables”](#) on page 10.

**Example**

`:CORR:FLAT:STOR "New_Flat_data"`

The preceding example stores the current user flatness table entries in a file named "New\_Flat\_data".

**Key Entry**            **Store To File**

## **[[:STATe]]**

**Supported**            All Models

```
[[:SOURce]:CORRection[:STATe] ON|OFF|1|0  
[:SOURce]:CORRection[:STATe]?
```

This command toggles the application of user flatness corrections to the current signal generator power output.

### **Example**

```
:CORR OFF
```

The preceding example turns off correction data.

```
*RST                    0
```

**Key Entry**            **Flatness Off On**



## Frequency Subsystem ([:SOURce])

### :FREQuency:CENTer

**Supported** All with Option 007

```
[ :SOURce]:FREQuency:CENTer <val>[<unit>] | UP | DOWN
[:SOURce]:FREQuency:CENTer? [MAXimum|MINimum]
```

This command sets the center frequency for a ramp sweep. The center frequency symmetrically divides the selected frequency span and is coupled to the start and stop frequency settings. The frequency range and reset values are dependent on the signal generator model and option number.

The query returns the start and stop ramp frequencies if the optional MAXimum or MINimum are used.

**\*RST**                    *Option 520:* +2.0000000000000E+10  
                          *Option 532:* +3.2000000000000E+10  
                          *Option 540:* +4.0000000000000E+10  
                          *Option 544:* +4.4000000000000E+10  
                          *Option 550:* +5.0000000000000E+10  
                          *Option 567:* +7.0000000000000E+10

**Range**                    *Option 520:* 250kHz–20GHZ  
                          *Option 532:* 250kHz–32GHZ  
                          *Option 540:* 250kHz–40GHZ  
                          *Option 544:* 250kHz–44GHZ  
                          *Option 550:* 250kHz–50GHZ  
                          *Option 567:* 250kHz–70GHZ<sup>a</sup>

a.67–70 GHz performance not specified

### Example

```
:FREQ:CENT 15GHZ
```

The preceding example sets the center frequency for a ramp sweep to 15 GHz.

**Key Entry**                **Freq Center12**

## :FREQuency:CHANnels:BAND

**Supported**            All Models

```
[[:SOURce]:FREQuency:CHANnels:BAND  
NBASe|NMOBile|BPGSm|MPGSm|BEGSm|MEGSm|BRGSm|MRGSm|BDCS|MDCS|BPCS|MPCS|B450|GM450|  
B480|M480|B850|M850|B8|M8|B15|M15|B390|B420|B460|B915|M380|M410|M450|M870|PHS|DECT  
[:SOURce]:FREQuency:CHANnels:BAND?
```

This command sets the frequency of the signal generator by specifying a frequency channel band. The frequency channel state must be enabled for this command to work. See [“:FREQuency:CHANnels\[:STATe\]” on page 117](#).

NBASe	This choice selects Standard Base as the frequency band for NADC.
NMOBile	This choice selects Standard Mobile as the frequency band for NADC.
BPGSm	This choice selects P-Gsm 900 Base as the frequency band for GSM.
MPGSm	This choice selects P-Gsm 900 Mobile as the frequency band for GSM.
BEGSm	This choice selects E-Gsm 900 Base as the frequency band for GSM.
MEGSm	This choice selects E-Gsm 900 Mobile as the frequency band for GSM.
BRGSm	This choice selects R-Gsm 900 Base as the frequency band for GSM.
MRGSm	This choice selects R-Gsm 900 Mobile as the frequency band for GSM.
BDCS	This choice selects DCS 1800 Base as the frequency band for GSM.
MDCS	This choice selects DCS 1800 Mobile as the frequency band for GSM.
BPCS	This choice selects PCS 1900 Base as the frequency band for GSM.
MPCS	This choice selects PCS 1900 Mobile as the frequency band for GSM.
B450	This choice selects Gsm 450 Base as the frequency band for GSM.
GM450	This choice selects Gsm 450 Mobile as the frequency band for GSM.
B480	This choice selects Gsm 480 Base as the frequency band for GSM.
M480	This choice selects Gsm 480 Mobile as the frequency band for GSM.
B850	This choice selects Gsm 850 Base as the frequency band for GSM.
M850	This choice selects Gsm 850 Mobile as the frequency band for GSM.
B8	This choice selects 800MHz Base as the frequency band for PDC.
M8	This choice selects 800MHz Mobile as the frequency band for PDC.
B15	This choice selects 1500MHz Base as the frequency band for PDC.
M15	This choice selects 1500MHz Mobile as the frequency band for PDC.
B390	This choice selects Base 390–400 as the frequency band for TETRA.
B420	This choice selects Base 420–430 as the frequency band for TETRA.
B460	This choice selects Base 460–470 as the frequency band for TETRA.
B915	This choice selects Base 915–921 as the frequency band for TETRA.
M380	This choice selects Mobile 380–390 as the frequency band for TETRA.
M410	This choice selects Mobile 410–420 as the frequency band for TETRA.
M450	This choice selects Mobile 450–460 as the frequency band for TETRA.
M870	This choice selects Mobile 870–876 as the frequency band for TETRA.
PHS	This choice selects Standard PHS as the frequency band.
DECT	This choice selects Standard DECT as the frequency band.

**Example**

:FREQ:CHAN:BAND DECT

The preceding example sets the frequency band to standard DECT.

<b>*RST</b>	<b>BPGS</b>			
<b>Key Entry</b>	<b>P-GSM Base PCS Base</b>	<b>E-GSM Base GSM 450 Base</b>	<b>R-GSM Base GSM 480 Base</b>	<b>DCS Base GSM 850 Base</b>
	<b>NADC Base</b>	<b>800MHz Base</b>	<b>1500MHz Base</b>	
	<b>Tetra Base 390/400</b>	<b>Tetra Base 420/430</b>	<b>Tetra Base 460/470</b>	
	<b>Tetra Base 915/921</b>	<b>PHS Standard</b>	<b>DECT Standard</b>	
	<b>P-GSM Mobile</b>	<b>E-GSM Mobile</b>	<b>R-GSM Mobile</b>	<b>DCS Mobile</b>
	<b>PCS Mobile</b>	<b>GSM 450 Mobile</b>	<b>GSM 480 Mobile</b>	<b>GSM 850 Mobile</b>
	<b>NADC Mobile</b>	<b>800MHz Mobile</b>	<b>1500MHz Mobile</b>	
	<b>Tetra Mobile 380/390</b>	<b>Tetra Mobile 410/420</b>	<b>Tetra Mobile 450/460</b>	
	<b>Tetra Mobile 870/876</b>			

## **:FREQuency:CHANnels:NUMBer**

**Supported**            All Models

```
[ :SOURce ] :FREQuency:CHANnels:NUMBer <number>  
[ :SOURce ] :FREQuency:CHANnels:NUMBer?
```

This command sets the frequency of the signal generator by specifying a channel number of a given frequency band.

The channel band and channel state must be enabled for this command to work. Refer to “:FREQuency:CHANnels[:STATe]” on page 117.

### **Example**

```
:FREQ:CHAN:NUMB 24
```

The preceding example sets the channel number to 24 for the current band.

<b>*RST</b>	+1	
<b>Range</b>	P-GSM Base/Mobile:	1-24
	E-GSM and R-GSM Base/Mobile:	1-1023
	DCS Base/Mobile:	512-885
	PCS Base/Mobile:	512-900
	GSM-450 Base/Mobile:	259-293
	GSM-480 Base/Mobile:	306-340
	GSM-850 Base/Mobile:	128-251
	NADC Base/Mobile:	1-1023
	800MHz Base/Mobile:	0-640
	1500MHz Base/Mobile:	0-960
	TETRA 380/390 Mobile:	3600-4000
	TETRA 390/4000 Base:	3600-4000
	TETRA 410/420 Mobile:	800-1200
	TETRA 420/430 Base:	800-1200
	TETRA 460/470: 2400 through 2800	2400-2800
	TETRA 870/876 Mobile:	600-640
	TETRA 915/921 Base:	600-940
	PHS Standard:	1-255
	DECT Standard:	0-9

**Key Entry**            **Channel Number**

## **:FREQuency:CHANnels[:STATE]**

**Supported** All Models

```
[ :SOURce]:FREQuency:CHANnels[:STATE] ON|OFF|1|0
[:SOURce]:FREQuency:CHANnels[:STATE]?
```

This command enables or disables the frequency channel and band selection. The signal generator frequency will be set to the channel frequency when the state is on. To set frequency channel bands refer to “:FREQuency:CHANnels:BAND” on page 114.

### **Example**

```
:FREQ:CHAN ON
```

The preceding example turns on the frequency channel.

```
*RST 0
```

**Key Entry** **Freq Channels Off On**

## **:FREQuency:FIXed**

**Supported** All Models

```
[ :SOURce]:FREQuency:FIXed <val><unit>|UP|DOWN
[:SOURce]:FREQuency:FIXed?
```

This command sets the signal generator output frequency , or increments or decrements the current RF frequency setting.

<val> A frequency value.

UP Increases the current frequency setting by the value set with the :FREQuency[:CW]:STEP[:INCRement] command found on page 125. The front panel up arrow key performs the same function.

DOWN Decreases the current frequency setting by the value set with the :FREQuency[:CW]:STEP[:INCRement] command found on page 125. The front panel down arrow key performs the same function.

To set the frequency mode, see “:FREQuency:MODE” on page 119. For a listing of signal generator frequency and power specifications, refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 153.

### **Example**

```
:FREQ:FIX 10GHZ
```

The preceding example sets the signal generator frequency to 10 GHz.

```
*RST Option 520: +2.0000000000000E+10
      Option 532: +3.2000000000000E+10
      Option 540: +4.0000000000000E+10
      Option 544: +4.4000000000000E+10
      Option 550: +5.0000000000000E+10
      Option 567: +6.7000000000000E+10
```

**Range**                    **Option 520:** 250kHz–20GHZ  
Option 532: 250kHz–32GHZ  
Option 540: 250kHz–40GHZ  
Option 544: 250kHz–44GHZ  
Option 550: 250kHz–50GHZ  
Option 567: 250kHz–70GHz<sup>a</sup>

a. 67–70 GHz performance not specified

**Key Entry**                **Freq CW**

## **:FREQuency:MANual**

**Supported**                All with Option 007

```
[ :SOURce ]:FREQuency:MANual <val><unit>  
[ :SOURce ]:FREQuency:MANual?
```

This command sets the RF output frequency when performing a ramp sweep in manual mode. The frequency value selected must fall within the range of the current start and stop frequency settings.

Entering a value with this command has no effect unless manual sweep mode is on. Refer to [:SWEep:MODE](#) command for setting the mode.

The variable <val> is a numeric value. The <units> variable can be expressed in Hz, KHz, MHz, or GHz.

### **Example**

```
:FREQ:MAN 10GHZ
```

The preceding example sets the signal generator manual ramp sweep frequency to 10 GHz.

**\*RST**                    **Option 520:** +2.0000000000000E+10  
Option 532: +3.2000000000000E+10  
Option 540: +4.0000000000000E+10  
Option 544: +4.4000000000000E+10  
Option 550: +5.0000000000000E+10  
Option 567: +6.7000000000000E+10

**Range**                    **Option 520:** 250kHz–20GHZ  
Option 532: 250kHz–32GHZ  
Option 540: 250kHz–40GHZ  
Option 544: 250kHz–44GHZ  
Option 550: 250kHz–50GHZ  
Option 567: 250kHz–70GHz<sup>a</sup>

a. 67–70 GHz performance not specified

**Key Entry**                **Manual Freq**

## :FREQuency:MODE

**Supported** All Models

```
[ :SOURce ] :FREQuency:MODE FIXed | CW | SWEep | LIST
[ :SOURce ] :FREQuency:MODE?
```

This command sets the frequency mode of the signal generator.

**FIXed and CW** These choices are synonymous. Any currently running frequency sweeps are turned off, and the current CW frequency settings are used to control the output frequency.

To set the frequency in the CW frequency mode, see “:FREQuency[:CW]” on page 124.

To set the frequency in the fixed frequency mode, see “:FREQuency:FIXed” on page 117.

**SWEep** The effects of this choice are determined by the sweep generation type selected (refer to “:SWEep:GENeration” on page 136). In analog sweep generation, the ramp sweep frequency settings (start, stop, center, and span) control the output frequency. In step sweep generation, the current step sweep frequency settings control the output frequency. In both cases, this selection also activates the sweep. This choice is available with Option 007 only.

**LIST** This choice selects the swept frequency mode. If sweep triggering is set to immediate along with continuous sweep mode, executing the command starts the LIST or STEP frequency sweep.

---

**NOTE** To perform a frequency and amplitude sweep, you must also select LIST or SWEep as the power mode (see “:MODE” on page 148).

---

### Example

```
:FREQ:MODE LIST
```

The preceding example selects a list frequency sweep.

```
*RST CW
```

Key Entry	Freq CW	Sweep Type	Freq	Off	Freq & Ampl
-----------	---------	------------	------	-----	-------------

## :FREQuency:MULTIplier

**Supported** All Models

```
[ :SOURce ] :FREQuency:MULTIplier <val>
[ :SOURce ] :FREQuency:MULTIplier?
```

This command sets the multiplier for the signal generator carrier frequency. For any multiplier other than one, the MULT indicator is shown in the frequency area of the display. The multiplier value is used to multiply the signal generator’s displayed frequency. The true frequency remains constant. For example, if the signal generator frequency is 20 GHz and a multiplier of 3 is selected, the displayed frequency will be 60 GHz. This feature is useful when working with mixers and multipliers.

### Example

```
:FREQ:MULT 2
```

The preceding example sets the carrier multiplier to 2.

```
*RST          +1.00000000E+000
```

**Key Entry**           **Freq Multiplier**

### :FREQuency:OFFSet

**Supported**           All Models

```
[[:SOURce]:FREQuency:OFFSet <val><units>  
[:SOURce]:FREQuency:OFFSet?
```

This command sets the frequency offset. When an offset has been entered, the OFFS indicator appears in the frequency area of the signal generator's front panel display and the frequency reading will include the offset value.

When any non-zero value is entered, the frequency offset state turns on; entering zero turns it off. To set the offset state independent of entering offset values refer to [:FREQuency:OFFSet:STATe](#) command.

### Example

```
:FREQ:OFFS 10GHZ
```

The preceding example sets the frequency offset to 10 GHz.

```
*RST          +0.00000000000000E+00
```

**Range**               -200 GHz to 200 GHz

**Key Entry**           **Freq Offset**

### :FREQuency:OFFSet:STATe

**Supported**           All Models

```
[[:SOURce]:FREQuency:OFFSet:STATe ON|OFF|1|0  
[:SOURce]:FREQuency:OFFSet:STATe?
```

This command enables or disables the offset frequency.

Entering OFF (0) will set the frequency offset to 0 Hz.

### Example

```
:FREQ:OFFS:STAT 0
```

The preceding example disables the frequency offset and sets the offset to 0 hertz.

```
*RST          0
```

**Key Entry**           **Freq Offset**



## **:FREQuency:REFeRence**

**Supported** All Models

```
[ :SOURce ]:FREQuency:REFeRence <val><units>
[ :SOURce ]:FREQuency:REFeRence?
```

This command sets the output reference frequency for the signal generator. Once the reference frequency is set, any change to the signal generator's CW frequency will be displayed referenced to 0 hertz. For example, if the signal generator's CW frequency is set to 100 megahertz and the frequency reference is set (the frequency reference state will automatically turn on). The frequency display will read 0 Hz. If you change the signal generator's CW frequency to 1 megahertz, the frequency display will read 1 megahertz. However, the true frequency is 101 megahertz. This can be verified by turning the frequency reference state off. The signal generator frequency display will read 101 megahertz. Refer to [:FREQuency:REFeRence:STATe](#) command.

### **Example**

```
:FREQ:REF 100MHz
```

The preceding example sets the output reference frequency to 100 megahertz.

```
*RST +0.00000000000000E+00
```

**Key Entry** **Freq Ref Set**

## **:FREQuency:REFeRence:SET**

**Supported** All Models

```
[ :SOURce ]:FREQuency:REFeRence:Set
```

This command sets the current CW output frequency, along with any offset, as a 0 hertz reference value.

```
*RST +0.00000000000000E+00
```

**Key Entry** **Freq Ref Set**

## **:FREQuency:REFeRence:STATe**

**Supported** All Models

```
[ :SOURce ]:FREQuency:REFeRence:STATe ON|OFF|1|0
[ :SOURce ]:FREQuency:REFeRence:STATe?
```

This command enables or disables the frequency reference mode. When the frequency reference mode is on, changes in the signal generator's CW frequency are displayed relative to the 0 hertz frequency reference. When the state is off, the front panel display indicates the true signal generator frequency.

### **Example**

```
:FREQ:REF:STAT OFF
```

The preceding example turns off the reference frequency mode.

```
*RST 0
```

**Key Entry** **Freq Ref Off On**

## :FREQuency:SPAN

**Supported** All with Option 007

```
[ :SOURce]:FREQuency:SPAN <num>[<freq_suffix>] |UP|DOWN  
[:SOURce]:FREQuency:SPAN? [MAXimum|MINimum]
```

This command sets the length of the frequency range for a ramp sweep. Span setting is symmetrically divided by the selected center frequency and is coupled to the start and stop frequency settings. The span range is dependent on the signal generator model and option number.

### Example

```
:FREQ:SPAN 100MHZ
```

The preceding example sets the frequency span to 100 megahertz.

```
*RST +0.00000000000000E+00
```

**Key Entry** **Freq Span**

## :FREQuency:STARt

**Supported** All Models

```
[ :SOURce]:FREQuency:STARt <val><units>  
[:SOURce]:FREQuency:STARt?
```

This command sets the frequency start point for a step sweep or ramp sweep (Option 007). In a ramp sweep setup, the selected value must be less than or equal to the value selected for the frequency stop point. In ramp sweep, this setting is coupled with the span and center frequency settings.

Refer to [\[:LEVel\]\[:IMMediate\]\[:AMPLitude\]](#) command for frequency and power specifications for different signal generator options and model numbers.

### Example

```
:FREQ:STAR 1GHZ
```

The preceding example sets the start frequency for a sweep to 1 GHz.

```
*RST Option 520: +2.0000000000000E+10  
Option 532: +3.2000000000000E+10  
Option 540: +4.0000000000000E+10  
Option 544: +4.4000000000000E+10  
Option 550: +5.0000000000000E+10  
Option 567: +6.7000000000000E+10
```

**Range**                    *Option 520:* 250kHz–20GHZ  
                               Option 532: 250kHz–32GHZ  
                               Option 540: 250kHz–40GHZ  
                               Option 544: 250kHz–44GHZ  
                               Option 550: 250kHz–50GHZ  
                               Option 567: 250kHz–70GHZ<sup>a</sup>

a.67–70 GHz performance not specified

**Key Entry**                **Freq Start**

### **:FREQUENCY:STOP**

**Supported**                All Models

```
[ :SOURce ]:FREQuency:STOP <val><units>
[ :SOURce ]:FREQuency:STOP?
```

This command sets the stop frequency for a step sweep or ramp sweep (Option 007). In a ramp sweep setup, the selected value must be greater than or equal to the value selected for the frequency start point. In ramp sweep, this setting is coupled with the span and center frequency settings.

Refer to [:LEVel][:IMMediate][:AMPLitude] command for frequency and power specifications for different signal generator options and model numbers.

### **Example**

```
:FREQ:STOP 10GHZ
```

The preceding example sets the stop frequency for a sweep to 10 GHz.

**\*RST**                    *Option 520:* +2.0000000000000E+10  
                               Option 532: +3.2000000000000E+10  
                               Option 540: +4.0000000000000E+10  
                               Option 544: +4.4000000000000E+10  
                               Option 550: +5.0000000000000E+10  
                               Option 567: +6.7000000000000E+10

**Range**                    *Option 520:* 250kHz–20GHZ  
                               Option 532: 250kHz–32GHZ  
                               Option 540: 250kHz–40GHZ  
                               Option 544: 250kHz–44GHZ  
                               Option 550: 250kHz–50GHZ  
                               Option 567: 250kHz–70GHZ<sup>a</sup>

a.67–70 GHz performance not specified

**Key Entry**                **Freq Stop**

## **:FREQuency[:CW]**

**Supported**            All Models

```
[ :SOURce]:FREQuency[:CW] <val><unit>|UP|DOWN  
[:SOURce]:FREQuency[:CW]?
```

This command sets the signal generator output frequency for the CW frequency mode, or increments or decrements the current RF frequency setting.

<val>                    A frequency value.

UP                        Increases the current frequency setting by the value set with the [:FREQuency\[:CW\]:STEP\[:INCRement\]](#) command found on [page 125](#). The front panel up arrow key performs the same function.

DOWN                     Decreases the current frequency setting by the value set with the [:FREQuency\[:CW\]:STEP\[:INCRement\]](#) command found on [page 125](#). The front panel down arrow key performs the same function.

To set the frequency mode to CW, refer to “[:FREQuency:MODE](#)” on [page 119](#).

### **Example**

```
:FREQ 12GHZ
```

The preceding example sets signal generator’s output frequency to 12 GHz.

**\*RST**                    **Option 520:** +2.0000000000000E+10  
Option 532: +3.2000000000000E+10  
Option 540: +4.0000000000000E+10  
Option 544: +4.4000000000000E+10  
Option 550: +5.0000000000000E+10  
Option 567: +6.7000000000000E+10

**Range**                    **Option 520:** 250kHz–20GHZ  
Option 532: 250kHz–32GHZ  
Option 540: 250kHz–40GHZ  
Option 544: 250kHz–44GHZ  
Option 550: 250kH–50GHZ  
Option 567: 250kHz–70GHZ<sup>a</sup>

a. 67–70 GHz performance not specified

**Key Entry**            **Frequency**

## **:FREQuency[:CW]:STEP[:INCRement]**

**Supported** All Models

```
[ :SOURce ]:FREQuency[:CW]:STEP[:INCRement] <val><unit>
[ :SOURce ]:FREQuency[:CW]:STEP[:INCRement]?
```

This command sets the incremental step value for the frequency parameter. The value set with this command is not affected by \*RST or a power cycle.

**Range** .01 Hz to 99 GHz

**Key Entry** **Incr Set**

## **:FREQuency:SYNThesis**

**Supported** All Models with Option UNY

```
[ :SOURce ]:FREQuency:SYNThesis 1|2
[ :SOURce ]:FREQuency:SYNThesis?
```

This command sets the phase-lock loop (PLL) bandwidth to optimize phase noise for offsets above and below 150 kHz.

- 1 This choice will select mode 1, which optimizes phase noise at offsets below 150 kHz.
- 2 This choice will select mode 2, which optimizes phase noise at offsets above 150 kHz.

**\*RST** +1

**Key Entry** **Optimize Phase Noise**

## **:PHASe:REFerence**

**Supported** All Models

```
[ :SOURce ]:PHASe:REFerence
```

This command sets the output phase reference to zero. Subsequent phase adjustments are set relative to the new reference.

**Key Entry** **Phase Ref Set**

## **:PHASe[:ADJust]**

**Supported** All Models

```
[ :SOURce ]:PHASe[:ADJust] <val><unit>
[ :SOURce ]:PHASe[:ADJust]?
```

This command adjusts the phase of the modulating signal. The query returns values in radians.

### **Example**

```
:PHAS 30DEG
```

The preceding example sets the phase of the modulating signal to 30 degrees relative to the previous phase setting.

**\*RST**                    +0.00000000E+000  
**Range**                 *Radians:* -3.14 to 3.14RAD     *Degrees:* -180 to 179DEG  
**Key Entry**            **Adjust Phase**

### **:ROSCillator:BANDwidth:DEFaults**

**Supported**            All Models with Option UNR/UNX/UNY

[ :SOURce]:ROSCillator:BANDwidth:DEFaults

This command resets the bandwidth of the reference oscillator to the factory-defined default state. The default value for the internal reference bandwidth is 125 Hz. The default value for the external reference bandwidth is listed below.

Option	Internal Reference	External Reference
UNR	125 Hz	25 Hz
UNX	125 Hz	650 Hz
UNY	125 Hz	650 Hz

**Key Entry**            **Restore Factory Defaults**

### **:ROSCillator:BANDwidth:EXTernal**

**Supported**            All Models with Option UNR/UNX/UNY

[ :SOURce]:ROSCillator:BANDwidth:EXTernal 25HZ|55HZ|125HZ|300HZ|650HZ  
[:SOURce]:ROSCillator:BANDwidth:EXTernal?

This command sets the bandwidth of the internal reference oscillator when an external reference is applied.

#### **Example**

:ROSC:BAND:EXT 300HZ

The preceding example sets the bandwidth of the internal oscillator to 300 hertz.

**Key Entry**            **External Ref Bandwidth**

### **:ROSCillator:BANDwidth:INTernal**

**Supported**            All Models with Option UNR/UNX/UNY

[ :SOURce]:ROSCillator:BANDwidth:INTernal 25HZ|55HZ|125HZ|300HZ|650HZ  
[:SOURce]:ROSCillator:BANDwidth:INTernal?

This command sets the bandwidth of the internal reference oscillator.

#### **Example**

```
:ROSC:BAND:INT 125HZ
```

The preceding example sets the bandwidth of the internal oscillator to 125 hertz.

**Key Entry**            **Internal Ref Bandwidth**

### **:ROSCillator:SOURce:AUTO**

**Supported**            All Models without Option UNR/UNX/UNY

```
[ :SOURce]:ROSCillator:SOURce:AUTO ON|OFF|1|0  

[:SOURce]:ROSCillator:SOURce:AUTO?
```

This command enables or disables the ability of the signal generator to automatically select between the internal and an external reference oscillator.

ON (1)                This choice enables the signal generator to detect when a valid reference signal is present at the 10 MHz IN connector and automatically switches from internal to external frequency reference.

OFF (0)              This choice selects the internal reference oscillator and disables the switching capability between the internal and an external frequency reference.

#### **Example**

```
:ROSC:SOUR:AUTO 0
```

The preceding example turns off the automatic selection of internal or external reference oscillators.

```
*RST                    1
```

**Key Entry**            **Ref Oscillator Source Auto Off On**

### **:ROSCillator:SOURce**

**Supported**            All Models

```
[ :SOURce]:ROSCillator:SOURce?
```

This command queries the reference oscillator source: INT (internal) or EXT (external).

## List/Sweep Subsystem ([:SOURce])

A complete sweep setup requires commands from other subsystems. [Table 3-1](#) shows the function and location of these commands.

**Table 3-1 Location of Commands from the other Subsystems**

Sweep Type	Function	Command Location	Key Entry under Sweep/List key
List and Step	Start/stop frequency sweep	“:FREQUency:MODE” (page 119)	<b>Freq Off</b>
	Start/stop amplitude sweep	“:MODE” (page 148)	<b>Ampl Off</b>
	Start/stop frequency and amplitude sweep <sup>a</sup>	“:MODE” (page 148) “:FREQUency:MODE” (page 119)	<b>Freq &amp; Ampl Off</b>
	Set up & control sweep triggering <sup>b</sup>	“Trigger Sweep Subsystem ([:SOURce])” (page 155)	See the <a href="#">Trigger Sweep Subsystem ([:SOURce])</a>
Step	Start frequency sweep	“:FREQUency:START” (page 122)	<b>Freq Start</b>
	Stop frequency sweep	“:FREQUency:STOP” (page 123)	<b>Freq Stop</b>
	Start amplitude sweep	“:START” (page 152)	<b>Ampl Start</b>
	Stop amplitude sweep	“:STOP” (page 152)	<b>Ampl Stop</b>

a. Execute both commands to start or stop a frequency and amplitude sweep.

b. For point to point triggering, see “:LIST:TRIGger:SOURce” on [page 133](#).

### :LIST:DIRrection

**Supported** All Models

```
[ :SOURce]:LIST:DIRrection UP|DOWN
```

```
[ :SOURce]:LIST:DIRrection?
```

This command sets the direction of a list or step sweep.

**UP** This choice enables a sweep in an ascending order:

- first to last point for a list sweep
- start to stop for a step sweep

**DOWN** This choice reverses the direction of the sweep.

#### Example

```
:LIST:DIR UP
```

The preceding example selects an ascending sweep direction.

```
*RST UP
```

**Key Entry** Sweep Direction Down Up



## :LIST:DWELI

**Supported** All Models

```
[ :SOURce ]:LIST:DWELl <val>{ ,<val>}
[ :SOURce ]:LIST:DWELl?
```

This command sets the dwell time for points in the current list sweep.

The variable <val> is expressed in units of seconds with a 0.001 resolution. If only one point is specified, that value is used for all points in the list. Otherwise, there must be a dwell point for each frequency and amplitude point in the list.

---

**NOTE** The dwell time <val> does not begin until the signal generator frequency and/or amplitude change has settled.

---

Dwell time is used when IMMEDIATE is the trigger source. Refer to :LIST:TRIGger:SOURce command for the trigger setting.

The dwell time is the amount of time the sweep pauses after setting the frequency and/or power for the current point.

The setting enabled by this command is not affected by a signal generator power cycle, preset, or \*RST command.

### Example

```
:LIST:DWEL .1, .2, .1, .2, .3
```

The preceding example sets the dwell time for a list of five points.

**Range** 0.001 to 60 seconds

## :LIST:DWELI:POINTS

**Supported** All Models

```
[ :SOURce ]:LIST:DWELl:POINTs?
```

This command queries the signal generator for the number of dwell points in the list sweep file.

## :LIST:DWELI:TYPE

**Supported** All Models

```
[ :SOURce ]:LIST:DWELl:TYPE LIST|STEP
[ :SOURce ]:LIST:DWELl:TYPE?
```

This command toggles the dwell time for the list sweep points between the values defined in the list sweep and the value for the step sweep.

**LIST** This choice selects the dwell times from the list sweep. Refer to :LIST:DWELI command for setting the list dwell points.

**STEP** This choice selects the dwell time from the step sweep. Refer to :SWEep:DWELI command for setting the step dwell.

**Example**

```
:LIST:DWEL:TYPE STEP
```

The preceding example selects the dwell time from step sweep values.

**\*RST**                   LIST

**Key Entry**           **Dwell Type List Step**

**:LIST:FREQuency**

**Supported**           All Models

```
[ :SOURce ]:LIST:FREQuency <val> { , <val> }  
[ :SOURce ]:LIST:FREQuency?
```

This command sets the frequency values for the current list sweep points. The maximum number of points is 3201. The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

The variable <val> is expressed in hertz.

For signal generator frequency and power specifications, refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 153.

**Example**

```
:LIST:FREQ 10GHZ,12GHZ,14GHZ,16GHZ
```

The preceding example sets the frequency value for a four point sweep.

**\*RST**                   **Option 520:** +2.0000000000000E+10  
                         **Option 532:** +3.2000000000000E+10  
                         **Option 540:** +4.0000000000000E+10  
                         **Option 544:** +4.4000000000000E+10  
                         **Option 550:** +5.0000000000000E+10  
                         **Option 567:** +6.7000000000000E+10

**Range**               **Option 520:** 250kHz–20GHZ  
                         **Option 532:** 250kHz–32GHZ  
                         **Option 540:** 250kHz–40GHZ  
                         **Option 544:** 250kHz–44GHZ  
                         **Option 550:** 250kHz–50GHZ  
                         **Option 567:** 250kHz–70GHz<sup>a</sup>

a. 67–70 GHz performance not specified

## **:LIST:FREQuency:POINts**

**Supported** All Models

```
[ :SOURce ] :LIST:FREQuency:POINts?
```

This command queries the current list sweep file for the number of frequency points.

## **:LIST:MANual**

**Supported** All Models

```
[ :SOURce ] :LIST:MANual <val> |UP|DOWN
```

```
[ :SOURce ] :LIST:MANual?
```

This command selects a list point or step sweep point as the current sweep point controlling the frequency and power output. If list or step mode is controlling frequency or power, or both, the indexed point in the respective list(s) is used. The **MANual** mode must be selected and sweep enabled for this command to have an effect. For information on setting the proper mode, see the [:LIST:MODE](#) command.

If the point selected is beyond the length of the longest enabled list, the point sets to the maximum possible point, and an error is generated.

### **Example**

```
:LIST:MAN UP
```

The preceding example selects the next positive direction, sequential point in the list.

**Range** List Sweep: 1 to 3201  
Step Sweep: 1 to 65535

**Key Entry** **Manual Point**

## **:LIST:MODE**

**Supported** All Models

```
[ :SOURce ] :LIST:MODE AUTO|MANual
```

```
[ :SOURce ] :LIST:MODE?
```

This command sets the operating mode for the current list or step sweep.

**AUTO** This choice enables the selected sweep type to perform a sweep of all points.

**MANual** This choice enables you to select an individual sweep point to control the RF output parameters. For more about selecting a sweep point, see [“:LIST:MANual” on page 131](#).

### **Example**

```
:LIST:MODE AUTO
```

The preceding example sets the mode to automatic.

**\*RST** AUTO

**Key Entry**            **Manual Mode Off On**

## **:LIST:POWer**

**Supported**            All Models

```
[ :SOURce ] :LIST:POWer <val> { , <val> }  
[ :SOURce ] :LIST:POWer ?
```

This command sets the amplitude for the current list sweep points.

The setting enabled by this command is not affected by signal generator power-on, preset, or \*RST.

During an amplitude sweep operation, signal generators with Option 1E1 protect the step attenuator by automatically switching to attenuator hold mode (OFF). The attenuator locks at its current setting and the amplitude sweep range is limited to 40 dB. The maximum number of points is 3201.

### **Example**

```
:LIST:POW .1, .2, .1, .3, .1, -.1
```

The preceding example sets the power level for a six point sweep list.

**Range**                See also: “[:LEVel][:IMMediate][:AMPLitude]” on page 153.

## **:LIST:POWer:POINts**

**Supported**            All Models

```
[ :SOURce ] :LIST:POWer:POINts ?
```

This command queries the number of power points in the current list sweep file.

## **:LIST:RETRace**

**Supported**            All Models

```
[ :SOURce ] :LIST:RETRace ON | OFF | 1 | 0  
[ :SOURce ] :LIST:RETRace ?
```

Upon completion of a single sweep operation, this command either resets the sweep to the first sweep point, or leaves it at the last sweep point. The command is valid for the list, step, or ramp (Option 007) single-sweep modes.

ON (1)                The sweep resets to the first sweep point.

OFF (0)               The sweep stays at the last sweep point.

### **Example**

```
:LIST:RETR 1
```

The preceding example sets the retrace on. The sweep will reset to the first point after completing a sweep.

**\*RST**                    1

**Key Entry**            **Sweep Retrace Off On**

## :LIST:TRIGger:SOURce

**Supported** All Models

```
[ :SOURce ] :LIST:TRIGger:SOURce BUS | IMMEDIATE | EXTERNAL | KEY
[ :SOURce ] :LIST:TRIGger:SOURce?
```

This command sets the trigger source for a list or step sweep event.

To set the sweep trigger, see “:TRIGger[:SEQUENCE]:SOURce” on page 106.

BUS	This choice enables GPIB triggering using the *TRG or GET command, or LAN and RS-232 triggering using the *TRG command.
IMMEDIATE	This choice enables immediate triggering of the sweep event.
EXTERNAL	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables triggering by pressing the front panel <b>Trigger</b> hardkey.

### Example

```
:LIST:TRIG:SOUR BUS
```

The preceding example sets the trigger source to the instrument BUS.

<b>*RST</b>	IMM			
<b>Key Entry</b>	<b>Bus</b>	<b>Free Run</b>	<b>Ext</b>	<b>Trigger Key</b>

## :LIST:TYPE

**Supported** All Models

```
[ :SOURce ] :LIST:TYPE LIST | STEP
[ :SOURce ] :LIST:TYPE?
```

This command selects the sweep type.

LIST	This type of sweep has arbitrary frequencies and amplitudes.
STEP	This type of sweep has equally spaced frequencies and amplitudes.

### Example

```
:LIST:TYPE LIST
```

The preceding example selects list as the sweep type.

<b>*RST</b>	STEP
<b>Key Entry</b>	<b>Sweep Type List Step</b>

## **:LIST:TYPE:LIST:INITialize:FSTep**

**Supported**            All Models

---

**CAUTION**     When you execute this command, the current list sweep data is overwritten. If needed, save the current data. For information on storing list sweep files, see “:STORe:LIST” on page 58.

---

[ :SOURce ] :LIST:TYPE:LIST:INITialize:FSTep

This command replaces the loaded list sweep data with the settings from the current step sweep data points. You can have only one sweep list at a time.

The maximum number of list sweep points is 1,601. When copying the step sweep settings over to a list sweep, ensure that the number of points in the step sweep do not exceed the maximum list sweep points.

**Key Entry**            **Load List From Step Sweep**

## **:LIST:TYPE:LIST:INITialize:PRESet**

**Supported**            All Models

---

**CAUTION**     When you execute this command, the current list sweep data is overwritten. If needed, save the current data. For information on storing list sweep files, see “:STORe:LIST” on page 58.

---

[ :SOURce ] :LIST:TYPE:LIST:INITialize:PRESet

This command replaces the current list sweep data with a factory-defined file consisting of one point at a frequency, amplitude, and dwell time.

**Key Entry**            **Preset List**

## **:SWEep:CONTRol:STATe**

**Supported**            All models with Option 007

[ :SOURce ] :SWEep:CONTRol:STATe ON|OFF|1|0

[ :SOURce ] :SWEep:CONTRol:STATe?

This command sets the sweep control state for a PSG in a dual-PSG ramp sweep setup. When the sweep control is on, you can designate whether the PSG is operating as the master or the slave. For information on setting master and slave designations, see “:SWEep:CONTRol:TYPE” on page 135.

The dual-PSG ramp sweep setup uses a serial cable to connect the two signal generators. This connection enables one PSG to function as the master so that sweep, bandcross, and retrace times are synchronized between the two. Each PSG can have a different sweep range, but they must have identical sweep time settings.

### **Example**

:SWE:CONT:STAT 1

The preceding example sets the sweep control state to on.

```
*RST          0
Key Entry     Sweep Control
```

### **:SWEep:CONTRol:TYPE**

**Supported** All models with Option 007

```
[ :SOURce ] :SWEep:CONTRol:TYPE MASTER|SLAVE
[ :SOURce ] :SWEep:CONTRol:TYPE?
```

In a dual-PSG ramp sweep setup, this command designates whether the PSG is performing as the master or the slave. The master/slave setup requires two signal generators from the same instrument family. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* for more information.

**MASTer** This choice enables the PSG to provide the triggering.

**SLAVe** This choice causes the PSG to submit to the triggering parameters provided by the master PSG. You must set the slave PSG triggering to **CONTinuous** using the **:INITiate:CONTinuous[:ALL]** command.

### **Example**

```
:SWE:CONT:TYPE MAST
```

The preceding example sets the PSG as the master sweep control instrument.

```
*RST          0
Key Entry     Master or Slave
```

## :SWEep:DWELl

**Supported** All Models

[ :SOURce ] :SWEep:DWELl <val>

[ :SOURce ] :SWEep:DWELl?

This command enables you to set the dwell time for a step sweep.

The variable <val> is expressed in seconds with a 0.001 resolution.

---

**NOTE** The dwell time <val> does not begin until the signal generator has settled for the current frequency and/or amplitude change.

---

Dwell time is used when the trigger source is set to IMMEDIATE.

For the trigger setting, refer to “:LIST:TRIGger:SOURce” on page 133.

The dwell time is the amount of time the sweep pauses after setting the frequency or power, or both, for the current point.

### Example

```
:SWE:DWEL .1
```

The preceding example sets the dwell time for a step sweep to 100 milliseconds.

\*RST +2.00000000E-003

**Range** 0.001 ms to 60 seconds

**Key Entry** Step Dwell

## :SWEep:GENeration

**Supported** All models with Option 007

[ :SOURce ] :SWEep:GENeration ANALog | STEPped

[ :SOURce ] :SWEep:GENeration?

This command sets the sweep type to analog or stepped.

ANALog This choice selects a ramp sweep.

STEPped This choice selects a step sweep.

### Example

```
:SWE:GEN STEP
```

The preceding example selects a step sweep.

\*RST STEP

**Key Entry** Sweep Type



## **:SWEep:MODE**

**Supported** All models with Option 007

[ :SOURce ] :SWEep:MODE AUTO|MANual

[ :SOURce ] :SWEep:MODE?

This command sets the current ramp sweep operating mode.

**AUTO** This choice enables the signal generator to automatically sweep through the selected frequency range.

**MANual** This choice enables you to select a single frequency value within the current sweep range to control the RF output. For information on selecting the frequency value, see “:FREQuency:MANual” on page 118.

### **Example**

```
:SWE:MODE AUTO
```

The preceding example sets the signal generator to automatically complete a sweep.

```
*RST AUTO
```

**Key Entry** Manual Mode Off On

## **:SWEep:POINts**

**Supported** All Models

[ :SOURce ] :SWEep:POINts <val>

[ :SOURce ] :SWEep:POINts?

This command enables you to define the number of points in a step sweep.

### **Example**

```
:SWE:POIN 2001
```

The preceding example sets the number of step sweep points to 2001.

```
*RST 2
```

**Range** 2 to 65535

**Key Entry** # Points

## :SWEep:TIME

**Supported** All models with Option 007

```
[ :SOURce ] :SWEep:TIME <val><units>
```

```
[ :SOURce ] :SWEep:TIME?
```

This command enables you to set the sweep time for a ramp sweep in seconds. If this command is executed while the signal generator is in automatic sweep time mode, the manual sweep time mode is activated and the new sweep time value is applied. The sweep time cannot be set to a value faster than what the automatic mode provides.

The sweep time is the duration of the sweep from the start frequency to the stop frequency. It does not include the bandcross time that occurs during a sweep or the retrace time that occurs between sweep repetitions.

### Example

```
:SWE:TIME .250
```

The preceding example sets the ramp sweep time to 250 milliseconds.

**\*RST** 1.00000000E-002

**Range** 10 ms to 99 seconds

**Key Entry** Sweep Time

## :SWEep:TIME:AUTO

**Supported** All models with Option 007

```
[ :SOURce ] :SWEep:TIME:AUTO ON|OFF|0|1
```

```
[ :SOURce ] :SWEep:TIME:AUTO?
```

This command enables you to set the sweep time mode for a ramp sweep.

The sweep time is the duration of the sweep from the start frequency to the stop frequency. It does not include the bandcross time that occurs during a sweep or the retrace time that occurs between sweep repetitions.

ON (1) This choice enables the signal generator to automatically calculate and set the fastest allowable sweep time.

OFF (0) This choice enables you to select the sweep time. The sweep time cannot be set to a value faster than what the automatic mode provides. To set the sweep time refer to “:SWEep:TIME” on page 138.

### Example

```
:SWE:TIME:AUTO 0
```

The preceding example sets the ramp sweep time to manual allowing you to select a sweep time.

**\*RST** 1

**Key Entry** Sweep Time Manual Auto

## Marker Subsystem—Option 007 ([:SOURce])

### :MARKer:AMPLitude[:STATe]

**Supported** All models with Option 007

```
[ :SOURce]:MARKer:AMPLitude[:STATe] ON|OFF|1|0
[:SOURce]:MARKer:AMPLitude[:STATe]?
```

This command sets the amplitude marker state for the currently activated markers. When the state is switched on, the RF output signal exhibits a spike with a magnitude relative to the power level at each marker's set frequency. (To set the magnitude of the spike, refer to [:MARKer:AMPLitude:VALue](#) command.) The width of the amplitude spike is a nominal eight buckets, based on 1601 buckets per sweep.

#### Example

```
:MARK:AMPL ON
```

The preceding example enables amplitude markers.

```
*RST 0
```

**Key Entry** Amplitude Markers Off On

### :MARKer:AMPLitude:VALue

**Supported** All models with Option 007

```
[ :SOURce]:MARKer:AMPLitude:VALue <num>[DB]
[:SOURce]:MARKer:AMPLitude:VALue?
```

This command sets the relative power for the amplitude spikes at each marker's set frequency when the amplitude marker mode is activated. (To activate the amplitude markers, refer to [":MARKer:AMPLitude\[:STATe\]"](#) on page 139.)

#### Example

```
:MARK:AMPL:VAL 4DB
```

The preceding example sets the relative marker power to 4 dB for all markers.

```
*RST 2dB
```

**Range** -10 to +10 dB

**Key Entry** Marker Value

### :MARKer:AOFF

**Supported** All models with Option 007

```
[ :SOURce]:MARKer:AOFF
```

This command turns off all active markers.

**Key Entry** Turn Off Markers

## **:MARKer:DELta?**

**Supported** All models with Option 007

```
[ :SOURCE ]:MARKer:DELta? <num>, <num>
```

This query returns the frequency difference between two amplitude markers. The variables <num> are used to designate the marker numbers.

### **Example**

```
:MARK:DELt? 1,2
```

The preceding example returns the frequency difference between amplitude markers 1 and 2.

**Range** 0 to 9

## **:MARKer[0,1,2,3,4,5,6,7,8,9]:FREQuency**

**Supported** All models with Option 007

```
[ :SOURCE ]:MARKer[0,1,2,3,4,5,6,7,8,9]:FREQuency <val><unit>  
[ :SOURCE ]:MARKer[0,1,2,3,4,5,6,7,8,9]:FREQuency? MAXimum|MINimum
```

This command sets the frequency for a specific marker. If the marker designator [n] is not specified, marker 0 is the default. The frequency value must be within the current start, stop, frequency sweep range. Using the MAXimum or MINimum parameters in the query will return the frequency boundary values for the markers.

If the marker frequency mode is set to delta when the query is sent, the returned value is not absolute, but is relative to the reference marker. (See the [:MARKer:MODE](#) command for more information.)

### **Example**

```
:MARK2:FREQ 10GHZ
```

The preceding example places amplitude marker 2 at 10 GHz.

**\*RST** +5.25000000E+008

**Range** Equivalent to current sweep range

**Key Entry** **Marker Freq**

## **:MARKer:MODE**

**Supported** All Models with Option 007

```
[ :SOURCE ]:MARKer:MODE FREQuency|DELta  
[ :SOURCE ]:MARKer:MODE?
```

This command sets the frequency mode for all markers.

**FREQuency** The frequency values for the markers are absolute.

**DELta** The frequency values for the markers are relative to the designated reference marker. The reference marker must be designated before this mode is selected. (See the [:MARKer:REFerence](#) command to select a reference marker.)

### Example

```
:MARK:MODE DELT
```

The preceding example sets the marker mode to delta.

<b>*RST</b>	FREQuency
<b>Key Entry</b>	<b>Marker Delta Off On</b>

### :MARKer:REFerence

**Supported** All models with Option 007

```
[ :SOURCE]:MARKer:REFerence <marker>
```

```
[ :SOURCE]:MARKer:REFerence?
```

This command designates the reference marker when using markers in delta mode. The variable <marker> designates the marker number.

### Example

```
:MARK:REF 6
```

The preceding example sets marker 6 as the reference marker.

<b>*RST</b>	0
<b>Range</b>	0 to 9
<b>Key Entry</b>	<b>Delta Ref Set</b>

### :MARKer[0,1,2,3,4,5,6,7,8,9][:STATe]

**Supported** All models with Option 007

```
[ :SOURCE]:MARKer[0,1,2,3,4,5,6,7,8,9][:STATe] ON|OFF|1|0
```

```
[ :SOURCE]:MARKer[0,1,2,3,4,5,6,7,8,9][:STATe]?
```

This command turns a marker on or off. Marker 0 is the default if the marker designator [n] is not specified.

### Example

```
:MARK6 ON
```

The preceding example turns marker 6 on.

<b>*RST</b>	0
<b>Key Entry</b>	<b>Marker On Off</b>

## Power Subsystem ([:SOURce]:POWer)

### :ALC:BANDwidth | BWIDth

**Supported**            All Models

```
[ :SOURce]:POWer:ALC:BANDwidth|BWIDth <num>[<freq_suffix>]  
[:SOURce]:POWer:ALC:BANDwidth|BWIDth?
```

This command sets the bandwidth of the automatic leveling control (ALC) loop. You can select bandwidths of 100 Hz, 1 kHz, 10 kHz, or 100kHz. If you do not specify one of these exact bandwidths, your entry rounds to the nearest acceptable value. The bandwidth choices for this command are not effective if an internal I/Q source is being used. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* for information on ALC and bandwidth considerations.

#### Example

```
:POW:ALC:BWID 1KHZ
```

The preceding example sets the ALC bandwidth to 1 kHz.

```
*RST                    100.0
```

**Key Entry**            **ALC BW**

### :ALC:BANDwidth | BWIDth:AUTO

**Supported**            All Models

```
[ :SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO ON|OFF|1|0  
[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO?
```

This command sets the state of the automatic leveling control (ALC) automatic bandwidth function. When this state is turned on, the signal generator automatically selects the optimum bandwidth for the ALC.

#### Example

```
:POW:ALC:BWID:AUTO 0
```

The preceding example disables the automatic bandwidth optimizing function.

```
*RST                    1
```

**Key Entry**            **ALC BW**

## :ALC:LEVel

**Supported** E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce]:POWer:ALC:LEVel <value>DB
[:SOURce]:POWer:ALC:LEVel?
```

This command sets the automatic leveling control (ALC) level when the attenuator hold is active.

Use this command when the automatic attenuation mode is set to OFF (0). Refer to [:ATTenuation:AUTO](#) command for choosing the attenuator mode.

### Example

```
:POW:ALC:LEV 10DB
```

The preceding example sets the ALC to 10 dB.

```
*RST +1.00000000E+000
```

**Range** -20 to 25 dB

**Key Entry** **Set ALC Level**

## :ALC:SEARCh

**Supported** All Models

```
[ :SOURce]:POWer:ALC:SEARCh MANUAL|AUTO|SPAN
[:SOURce]:POWer:ALC:SEARCh?
```

Power Search is a cal routine which improves output power accuracy when ALC is off.

This command enables or disables the internal power search calibration. A power search is recommended for pulse-modulated signals with pulse widths less than one microsecond. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* for more information on ALC and the power search function.

Power search has three modes of operation:

Manual	This choice executes a single power search of the current RF output signal when the 'Do Power Search' key is pressed.
Auto	This choice executes the power search automatically with each change in RF frequency or power when the 'Do Power Search' key is pressed.
Span	The cal routine pre-computes the cal values over a range of frequencies and applies the values as the frequency changes. When the 'Do Power Search' key is pressed, the cal will pre-compute the cal values at all selected frequencies.  Selecting Span mode causes subsequent power searches done by pressing the 'Do Power Search' key to be performed over an entire range of frequencies at one time. The power search corrections are then stored and used whenever the signal generator is tuned within the selected range. Due to thermal drift, the power search values should be periodically recomputed.

Use this command when the automatic leveling control (ALC) state is set to OFF (0). Refer to [:ALC\[:STATe\]](#) command for setting the ALC state.

If ON was previously selected, executing ONCE will cause OFF to be the current selection after the

power search is completed.

**Example**

```
:POW:ALC:SEAR ONCE
```

The preceding example starts a single power search of the RF output signal.

```
*RST 0
```

Key Entry	Power Search Manual Auto Span	Do Power Search
-----------	-------------------------------	-----------------

**:ALC:SEARch:REFeRence**

**Supported** All Models

```
[[:SOURce]:POWer:ALC:SEARch:REFeRence RMS|FIXEd|MANual|MODulated  
[:SOURce]:POWer:ALC:SEARch:REFeRence?
```

The Power Search Reference allows selection of the power search reference voltage used while the RF signal is being modulated.

During power search the I/Q modulator is biased with a voltage equivalent to the RMS level of the active ARB waveform. The RMS level of the ARB waveform can be computed by the user and set in the waveform header file. If the RMS level is not set in the waveform header file, the value will be computed by the firmware when the waveform is selected and played.

When power search is complete, the I/Q modulator bias is removed and the internal ARB signal is re-applied to the I/Q modulator.

The reference can be:

- RMS** Uses a DC bias equivalent to the current I/Q data. This selection is used only when the internal baseband generator (ARB) is active.
- FIXEd** This selection is typically used with externally supplied I/Q modulating signals. During the power search, the I/Q modulation is disabled and the I/Q modulator is biased with a fixed 1.0 volt DC signal. This is equivalent to the specified external input signal level. When the power search is complete, the DC bias is removed and the I/Q signal is re-applied to the I/Q modulator.
- MANual RMS** This selection is typically used when the FIXEd level of 1.0 volt is not appropriate for the external I/Q signal level or when the internal RMS calculated value is not representative of the desired power level. This selection requires the user to specify the DC bias level applied to the I/Q modulator during the power search.
- MODulated** This selection can be used with internally generated or externally applied I/Q modulation. When MOD is selected, the actual modulating signal is used to provide an AC I/Q bias on the I/Q modulator. Due to the time varying nature of the I/Q data, this search can take longer to settle and be less accurate than the other reference bias types. This bias type is not recommended for use with bursted or low symbol rate modulation formats.

**Example**

```
:POW:ALC:SEAR:REF FIX
```

The preceding example selects a fixed voltage as the reference for a power search.



\*RST                      FIXed

---

**NOTE** When the internal baseband generator is enabled, the reference is automatically set to RMS.

---

**Key Entry                      Power Search Reference Fixed Mod**

### **:ALC:SEARch:REFerence:LEVel**

*This function is only active when the power search reference type is set to Manual.*

**Supported                      All Models**

```
[ :SOURce ] :POWer:ALC:SEARch:REF:LEVel
[ :SOURce ] :POWer:ALC:SEARch:REFerence:LEVel?
```

This command allows adjustment of the voltage level used as the power search reference when the reference type is set to MANual. Setting this voltage is equivalent to varying the waveform RMS voltage and is used to provide a DC bias on the I/Q modulator that approximates the level of the modulating voltage.

#### **Example**

```
:POW:ALC:SEAR:REF:LEV 0.51
```

The preceding example sets the power search reference level to .51 volts rms.

\*RST                      1.0 V

**Key Entry                      Power Search Reference Level**

### **:ALC:SEARch:SPAN:START**

**Supported                      All Models**

```
[ :SOURce ] :POWer:ALC:SEARch:SPAN:START <val><units>
[ :SOURce ] :POWer:ALC:SEARch:SPAN:START?
```

This command sets the start frequency for a power search over a user-defined range. The start frequency has no default value. The start frequency value will be set before powering off the instrument.

#### **Example**

```
:POW:ALC:SEAR:SPAN:START 12GHZ
```

The preceding example selects 12 GHz as the start frequency for a power search.

**Key Entry                      Start Frequency**

### **:ALC:SEARch:SPAN:STOP**

**Supported                      All Models**

```
[ :SOURce ] :POWer:ALC:SEARch:SPAN:STOP <val><units>
[ :SOURce ] :POWer:ALC:SEARch:SPAN:STOP?
```

This command sets the stop frequency for a power search over a user-defined range. The stop

frequency has no default value. The stop frequency value will be set before powering off the instrument

**Example**

```
:POW:ALC:SEAR:SPAN:STOP 20GHZ
```

The preceding example selects 20 GHz as the stop frequency for a power search.

**Key Entry**            **Stop Frequency**

**:ALC:SEARch:SPAN:TYPE FULL|USER**

**Supported**            All Models

```
[ :SOURce]:POWer:ALC:SEARch:SPAN:TYPE FULL|USER  
[:SOURce]:POWer:ALC:SEARch:SPAN:TYPE?
```

This command enables you to select the frequency range for a power search. You can specify the range (USER) or you can select the full range (FULL) of the signal generator.

**Example**

```
:POW:ALC:SEAR:SPAN:TYPE USER
```

The preceding example selects a user-defined frequency range for the power search.

**Key Entry**            **Span Type User Full**

**:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0**

**Supported**            All Models

```
[ :SOURce]:POWer:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0  
[:SOURce]:POWer:ALC:SEARch:SPAN[:STATe]?
```

This command enables (1) or disables (0) the span mode, allowing you to perform power searches over a selected range of frequencies. The power search corrections are then stored and used whenever the signal generator is tuned within the selected range.

**Example**

```
:POW:ALC:SEAR:SPAN ON
```

The preceding example enables the span mode.

**:ALC:SOURce**

**Supported**            All Models

```
[ :SOURce]:POWer:ALC:SOURce INTernal|DIODE|MMHead  
[:SOURce]:POWer:ALC:SOURce?
```

This command enables you to select an automatic level control (ALC) source. You can select the internal ALC source, an external detector source, or a millimeter-wave source module. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* for more information on ALC leveling, bandwidth, and the power search function.

**Example**

```
:POW:ALC:SOUR MMH
```

The preceding example selects an Agilent 8355x series external millimeter head as the source (the unit must be connected to the signal generator).

```
*RST INT
```

**Key Entry**            **Leveling Mode**

### **:ALC:SOURce:EXTernal:COUPling**

**Supported**            All Models

```
[ :SOURce ] : POWer : ALC : SOURce : EXTernal : COUPling <value>DB
[ :SOURce ] : POWer : ALC : SOURce : EXTernal : COUPling?
```

This command sets the external detector coupling factor. Use the [:ALC:SOURce](#) command when DIODE is the selected ALC source. (0 to 32 coupling value).

#### **Example**

```
:POW:ALC:SOUR:EXT:COUP 20DB
```

The preceding example sets the external coupling factor to 20 dB.

```
*RST +1.60000000E+001
```

**Range**                -200 to 200 dB.

**Key Entry**            **Ext Detector Coupling Factor**

### **:ALC[:STATe]**

**Supported**            All Models

```
[ :SOURce ] : POWer : ALC [ : STATe ] ON | OFF | 1 | 0
[ :SOURce ] : POWer : ALC [ : STATe ] ?
```

This command enables or disables the automatic leveling control (ALC) circuit. The purpose of the ALC circuit is to hold output power at a desired level by adjusting the signal generator power circuits for power drift. Power will drift over time and with changes in temperature. Refer to the *E8257D/67D, E8663D PSG Signal Generators User's Guide* for more information on the ALC.

#### **Example**

```
:POW:ALC ON
```

The preceding example sets the ALC on.

```
*RST 1
```

**Key Entry**            **ALC Off On**

### **:ATTenuation**

**Supported**            E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce ] : POWer : ATTenuation <val><unit>
[ :SOURce ] : POWer : ATTenuation?
```

This command sets the attenuation level when the attenuator hold is active. For the E8267D, the

attenuation is set in increments of 5 dB. For the E8257D/E8663D with Option 1E1, the progression is 0, 5, 15, 25 and continues in 10 dB increments. For the E8257D with Option 550 or Option 567, the attenuation is set in increments of 10 dB with a range of 0 to 90 dB.

The output power is the ALC level minus the attenuator setting.

Use this command when the automatic attenuation mode is set to OFF (0). Refer to [:ATTenuation:AUTO](#) command for choosing the attenuator mode.

### Example

```
:POW:ATT 10DB
```

The preceding example sets the attenuator to 10 dB.

```
*RST          +115
```

**Range** 0 to 115 dB

**Key Entry** **Set Atten**

### :ATTenuation:AUTO

**Supported** E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce]:POWer:ATTenuation:AUTO ON|OFF|1|0  
[:SOURce]:POWer:ATTenuation:AUTO?
```

This command sets the state of the attenuator hold function.

ON (1) This choice enables the attenuator to operate normally.

OFF (0) This choice holds the attenuator at its current setting or at a selected value that will not change during power adjustments.

OFF (0) eliminates the power discontinuity normally associated with the attenuator switching during power adjustments. During an amplitude sweep operation, signal generators with Option 1E1 protect the step attenuator by automatically switching to attenuator hold mode (ON). The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

### Example

```
:POW:ATT:AUTO OFF
```

The preceding example turns off the attenuator hold function.

```
*RST          1
```

**Key Entry** **Atten Hold Off On**

### :MODE

**Supported** All Models

```
[ :SOURce]:POWer:MODE FIXEd|SWEEp|LIST  
[:SOURce]:POWer:MODE?
```

This command starts or stops an amplitude sweep and sets the power mode of the signal generator.

FIXEd This choice stops a power sweep and allows the signal generator to operate at a fixed power level. Refer to [\[:LEVel\]\[:IMMediate\]\[:AMPLitude\]](#) command for more

information on running power sweeps and setting CW amplitude settings that control the output power.

SWEep	The effects of this choice are determined by the sweep generation type selected (refer to “:SWEep:GENeration” on page 136). If you are using analog sweep generation, the current ramp sweep amplitude settings (start and stop) control the output power. If you are using step sweep generation, the current step sweep amplitude settings control the output power. In both cases, this selection also activates the sweep. This choice is available with Option 007 only.
LIST	This choice selects the swept power mode. If sweep triggering is set to immediate along with continuous sweep mode, executing the command starts the LIST or STEP frequency sweep.

---

**NOTE** To perform a frequency and amplitude sweep, you must also select LIST or SWEep as the frequency mode (see “:FREQuency:MODE” on page 119).

---

**Example**

```
:POW:MODE LIST
```

The preceding example sets list as the amplitude sweep mode.

```
*RST FIX
```

Key Entry	Sweep Type	Ampl	Off	Freq & Ampl
-----------	------------	------	-----	-------------

**:NOISe[:STATe]**

**Supported** All Models

```
[ :SOURce ] :POWer :NOISe [ :STATe ] ON | OFF | 1 | 0  
[ :SOURce ] :POWer :NOISe [ :STATe ] ?
```

This command enables the optimize signal-to-noise ratio (SNR) state on or off. A front panel LED indicates an On state and optimizes the attenuator and ALC setting to provide optimal signal-to-noise performance; it does not change the RF output power.

---

**NOTE** This mode is mutually exclusive with attenuator hold (Atten Hold), and any modulation type. A settings conflict error will be generated if attenuator hold or any modulation is activated when optimize signal-to-noise is active (On).

---

**Example**

```
:POW:NOIS ON
```

The preceding example enables the SNR On.

**Default** Off

**Key Entry** Optimize S/N Off/On

## **:POWer:LIMit[:MAX]:ADJust**

**Supported** E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce ] :POWer:LIMit [ :MAX ] :ADJust <ON|OFF|1|0>
```

```
[ :SOURce ] :POWer:LIMit [ :MAX ] :ADJust?
```

This command locks or unlocks the RF output power limit adjust function. Refer to the *Users Guide* and to the *Key Reference*. See also “[:POWer:LIMit\[:MAX\]](#)” on page 150.

ON (1) Unlocks (sets to Adjust) the manual RF output power limit adjustment (i.e. The RF output limit can now be adjusted).

OFF (0) Locks the manual RF output power limit adjustment (i.e. The RF output limit cannot be adjusted). (This is the default.)

### **Example**

```
:POW:LIM:ADJ ON
```

The preceding example unlocks the power limit adjust function (i.e. sets Power Limit Adjust to Adjust).

```
*RST 0
```

**Key Entry**            **RF Output Limit Lock Adjust**

## **:POWer:LIMit[:MAX]**

**Supported** E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce ] :POWer:LIMit [ :MAX ] <amplitude>
```

```
[ :SOURce ] :POWer:LIMit [ :MAX ]?
```

This command sets the RF output power limit when the power limit adjust function has been set to Adjust. Refer to the *Users Guide* and to the *Key Reference*. See also “[:POWer:LIMit\[:MAX\]:ADJust](#)” on page 150.

### **Example**

```
:POW:LIM 30 dBm
```

The preceding example sets the signal generator to 30 dBm of output power.

```
*RST 25 dBm
```

**Key Entry**            **RF Output Limit**

## **:PROTection:STATe**

**Supported** E8257D/E8663D with Option 1E1 and E8267D

```
[ :SOURce ] :POWer:PROTection [ :STATe ] ON|OFF|1|0
```

```
[ :SOURce ] :POWer:PROTection [ :STATe ]?
```

This command enables or disables the power search protection function. The power search protection function sets the attenuator to its maximum level whenever a power search is initiated. This can be

used to protect devices that are sensitive to high average power or high power changes. The trade off on using the power protection function is decreased attenuator life, as the attenuator will switch to its maximum setting during a power search.

---

**NOTE** Continual or excessive use of the power search protection function can decrease attenuator life.

---

ON (1) Causes the attenuator to switch to and hold its maximum level setting during a power search.

OFF (0) Sets the attenuator normal mode. The attenuator is not used during power search.

**Example**

```
:POW:PROT ON
```

The preceding example enables the power inhibit function.

```
*RST 0
```

**Key Entry** **RF During Power Search Normal Minimum**

**:REference**

**Supported** All Models

```
[ :SOURce]:POWer:REference <val><unit>  
[:SOURce]:POWer:REference?
```

This command sets the power level for the signal generator RF output reference. The RF output power is referenced to the value entered in this command.

**Example**

```
:POW:REF 50DBM
```

The preceding example sets the RF output power reference to 50 dBm.

```
*RST +0.00000000E+000
```

**Range** -400 to 300 dBm

**Key Entry** **Ampl Ref Set**

**:REference:STATe**

**Supported** All Models

```
[ :SOURce]:POWer:REference:STATe ON|OFF|1|0  
[:SOURce]:POWer:REference:STATe?
```

This command enables or disables the RF output reference.

ON (1) Sets the power reference state ON. dB is the unit displayed for commands (“:ANNotation:AMPLitude:UNIT” on page 30 and “:POWer” on page 108).

OFF (0) Sets the power reference state OFF.

Once the reference state is ON, all subsequent output power settings are set relative to the reference value. Amplitude offsets can be used with the amplitude reference mode.

### Example

```
:POW:REF:STAT 1
```

The preceding example sets the reference state on.

```
*RST          0  
Key Entry    Ampl Ref Off On
```

### :START

**Supported** All Models

```
[ :SOURce ]:POWer:STARt <val><unit>  
[ :SOURce ]:POWer:STARt?
```

This command sets the amplitude of the first point in a step or ramp sweep (Option 007).

During an amplitude sweep operation, signal generators with Option 1E1 protect the step attenuator by automatically switching to attenuator hold (ON) mode. The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

### Example

```
:POW:STAR -30DBM
```

The preceding example sets the amplitude of the first point in the sweep to -30 dBm.

```
*RST          Depends on model and option number  
Range        Refer to [ :LEVel ][:IMMediate ][:AMPLitude ] command for the output power ranges.  
Key Entry    Ampl Start
```

### :STOP

**Supported** All Models

```
[ :SOURce ]:POWer:STOP <val><unit>  
[ :SOURce ]:POWer:STOP?
```

This command sets the amplitude of the last point in a step or ramp sweep (Option 007).

During an amplitude sweep, signal generators with Option 1E1 protect the step attenuator by switching to attenuator hold (ON) mode. The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

### Example

```
:POW:STOP -10DBM
```

The preceding example sets the amplitude of the last point in the sweep to -10 dBm.

```
*RST          Depends on model and option number.  
Range        See the [ :LEVel ][:IMMediate ][:AMPLitude ] command for the available power ranges.  
Key Entry    Ampl Stop
```



### **[:LEVel][:IMMediate]:OFFSet**

**Supported** All Models

```
[:SOURCE]:POWER[:LEVel][:IMMediate]:OFFSet <val><unit>
[:SOURCE]:POWER[:LEVel][:IMMediate]:OFFSet?
```

This command sets the power offset value as a dB power offset to the actual RF output. This simulates a power level at a test point beyond the RF OUTPUT connector without changing the actual RF output power. The offset value only affects the displayed amplitude setting.

You can enter an amplitude offset anytime in either normal operation or amplitude reference mode.

#### **Example**

```
:POW:OFFS 10DB
```

The preceding example sets the amplitude offset to 10 dB.

**\*RST** +0.00000000E+000

**Range** -200 to 200 dB

**Key Entry** **Ampl Offset**

### **[:LEVel][:IMMediate][:AMPLitude]**

**Supported** All Models

```
[:SOURCE]:POWER[:LEVel][:IMMediate][:AMPLitude] <val><unit>
[:SOURCE]:POWER[:LEVel][:IMMediate][:AMPLitude]?
```

This command sets the RF output power.

The ranges for this command are specified values from the data sheet.

#### **Example**

```
:POW 0DBM
```

The preceding example sets the signal generator output power level to 0 dBm.

**\*RST** Depends on model and option number

**Range** See data sheet

**Key Entry** **Amplitude**

## **[[:LEVel]][:IMMediate][:AMPLitude]:STEP[:INCRement]**

**Supported**            All Models

```
[ :SOURce]:POWer[ :LEVel ][ :IMMediate ][ :AMPLitude]:STEP[ :INCRement] <val><unit>  
[:SOURce]:POWer[ :LEVel ][ :IMMediate ][ :AMPLitude]:STEP[ :INCRement]?
```

This command sets the increment value for the output power/amplitude. The increment values are persistent.

The ranges for this command are specified values from the data sheet.

**\*RST**                    Value is persistent (Factory value is 1 dB).

**Range**                    .01 to 100 dB

**Key Entry**                **Incr Set**

## Trigger Sweep Subsystem ([:SOURce])

### :TSweep

**Supported**            All Models

[ :SOURce ] :TSweep

This command aborts the current sweep, then either arms or arms and starts a single list, step, or ramp sweep (Option 007), depending on the trigger type.

The command performs the following:

- arms a single sweep when BUS, EXTERNAL, or KEY is the trigger source selection
- arms and starts a single sweep when IMMEDIATE is the trigger source selection

**Key Entry**            **Single Sweep**



---

## 4 Analog Commands

This chapter provides SCPI descriptions for subsystems dedicated to the E8257D PSG Analog, E8663D PSG Analog and E8267D PSG Vector signal generators. The following is a list of the subsystems:

- “Amplitude Subsystem ([:SOURce])” on page 157
- “Frequency Modulation Subsystem ([:SOURce])” on page 168
- “Low Frequency Output Subsystem ([:SOURce]:LFOutput)” on page 176
- “Phase Modulation Subsystem ([:SOURce])” on page 181
- “Pulse Modulation Subsystem ([:SOURce])” on page 190

### Amplitude Subsystem ([:SOURce])

**:AM[1]|2...**

**Supported**            All Models

[[:SOURce]:AM[1]|2]...

This prefix enables the selection of the AM path and is part of most SCPI commands associated with this subsystem. The two paths are equivalent to the **AM Path 1 2** softkey.

AM1                    **AM Path 1 2** with 1 selected

AM2                    **AM Path 1 2** with 2 selected

When just AM is shown in a command, the command defaults to path 1.

Each path is set up separately. When a SCPI command uses AM1, only path one is affected. Consequently, when AM2 is selected, only path two is set up. However, the depth of the signals for the two paths can be coupled.

The two AM paths can be on at the same time provided the following conditions have been met:

- dual-sine or swept-sine is not one of the selections for the waveform type
- Each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)

## **:AM:INTernal:FREQuency:STEP[:INCRement]**

**Supported** All Models

```
[ :SOURce ] :AM:INTernal:FREQuency:STEP [ :INCRement ] <num> | MAXimum | MINimum | DEFault  
[ :SOURce ] :AM:INTernal:FREQuency:STEP [ :INCRement ] ?
```

This command sets the step value for the AM internal frequency.

The step value set by this command is used with the UP and DOWN choices for the [:AM\[1\]|2:INTernal\[1\]|2:FREQuency](#) command.

The step value set with this command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:AM:INT:FREQ:STEP 1E3
```

The preceding example sets the step size to 1000 hertz.

**Range** 0.5 to 1E6

**Key Entry** **Incr Set**

## **:AM:MODE**

**Supported** All models with Option UNT

```
[ :SOURce ] :AM:MODE DEEP | NORMal  
[ :SOURce ] :AM:MODE ?
```

This command sets the mode for amplitude modulation.

**DEEP** This choice enables amplitude modulation depth with a greater dynamic range than normal mode which utilizes the ALC. DEEP has no specified parameters and emulates the amplitude modulation NORMAL mode with the ALC disabled.

**NORMAL** This choice maintains the amplitude modulation standard behavior and has specified parameters as outlined in the data sheet.

The ALC is disabled when the carrier amplitude is less than -10 dBm and DEEP is the AM mode.

DEEP is limited to repetitive AM and will not work with a dc modulation signal.

### **Example**

```
:AM:MODE NORM
```

The preceding example selects the normal mode for amplitude modulation.

**\*RST** NORM

**Key Entry** **AM Mode Normal Deep**

## **:AM:POLarity**

**Supported**            Option UNT

```
[ :SOURce]:AM[1]|2:POLarity NORMal|INVerted
[:SOURce]:AM[1]|2:POLarity?
```

This command sets the AM polarity mode for the RF output amplitude when an AM input voltage is applied to one of the instrument's rear panel input connectors. Refer to the *User's Guide*.

**NORMal**                This choice configures the PSG so that a negative AM input voltage will decrease the RF output amplitude.

**INVerted**              This choice configures the PSG so that a positive AM input voltage will decrease the RF output amplitude.

**\*RST**                    NORM

**Key Entry**             **AM Polarity Norm Inv**

## **:AM:WIDeband:SENSitivity**

**Supported**            E8267D with Option UNT

```
[ :SOURce]:AM:WIDeband:SENSitivity <val>
[:SOURce]:AM:WIDeband:SENSitivity?
```

This command sets the sensitivity level of the wideband AM signal in units of dB/volt. Sensitivity is .5V = 100% and is linear with .25V = 50%. Wideband AM uses input from the front panel I INPUT.

### **Example**

```
:AM:WID:SENS 20
```

The preceding example sets the sensitivity level to 20%.

**\*RST**                    +2.00000000E+001

**Range**                    0 to 40 dB

**Key Entry**             **AM Depth**

## **:AM:WIDeband:STATe**

**Supported**            E8267D with Option UNT

```
[ :SOURce]:AM:WIDeband:STATe ON|OFF|1|0
[:SOURce]:AM:WIDeband:STATe?
```

This command enables or disables wideband amplitude modulation. The RF carrier is modulated when the instrument's [:MODulation\[:STATe\]](#) command is enabled ON. The signal generator's I input is used to drive wideband AM modulation.

Whenever wideband amplitude modulation is enabled, the AM annunciator appears on the signal generator's front panel display. Both AM paths (1 and 2) may be simultaneously set to ON by the [:AM\[1\]|2...](#) command.

### **Example**

```
:AM:WID:STAT 0
```

The preceding example turns off wideband amplitude modulation.

```
*RST 0
```

**Key Entry**            **AM Off On**

### **:AM[1]|2:EXternal[1]|2:COUpling**

**Supported**            All Models

```
[ :SOURce]:AM[1]|2:EXternal[1]|2:COUpling AC|DC
```

```
[ :SOURce]:AM[1]|2:EXternal[1]|2:COUpling?
```

This command sets the coupling type for the selected external input. The command does not change the active source or switch the modulation on or off. The modulating signal may be the sum of several signals, with either internal or external sources.

AC                      This choice will pass only ac signal components.

DC                      This choice will pass both ac and dc signal components.

#### **Example**

```
:AM1:EXT1:COUP AC
```

The preceding example sets the AM path 1, external 1 source coupling to AC.

```
*RST DC
```

**Key Entry**            **Ext Coupling DC AC**

### **:AM[1]|2:EXternal[1]|2:IMPedance**

**Supported**            All Models

```
[ :SOURce]:AM[1]|2:EXternal[1]|2:IMPedance <50|600>
```

```
[ :SOURce]:AM[1]|2:EXternal[1]|2:IMPedance?
```

This commands sets the impedance for the external input.

#### **Example**

```
:AM1:EXT1:IMP 600
```

The preceding example sets the AM path 1, external 1 source impedance to 600 ohms.

```
*RST +5.00000000E+001
```

**Key Entry**            **Ext Impedance 50 Ohm 600 Ohm**



## :AM[1]|2:INTernal[1]|2:FREQuency

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INTernal[1]|2:FREQuency <val><units>|UP|DOWN
[:SOURce]:AM[1]|2:INTernal[1]|2:FREQuency?
```

This command sets the internal AM rate using the variable <val><units>. The command, used with the UP|DOWN parameters, will change the frequency rate by a user-defined step value. For setting the value associated with the UP and DOWN choices, refer to the [:PULM:INTernal\[1\]:FREQuency:STEP](#) command.

The command changes:

- the frequency rate of the first tone of a dual-sine waveform
- the start frequency for a swept-sine waveform
- the AM frequency rate for all other waveforms

Refer to [:AM\[1\]|2:INTernal\[1\]|2:FUNCTion:SHAPE](#) command for the waveform selection.

### Example

```
:AM1:INT2:FREQ UP
```

The preceding example increases the modulation rate for AM path 1, and AM internal source 2 by the step value set with the [:AM:INTernal:FREQuency:STEP\[:INCRement\]](#) command.

```
*RST                +4.00000000E+002
Range               Dual-Sine & Sine: 0.5 Hz to 1 MHz    Swept-Sine: 1 Hz to 1MHz
                   All Other Waveforms: 0.5 Hz to 100 kHz
Key Entry          AM Tone 1 Rate    AM Start Rate    AM Rate
```

## :AM[1]|2:INTernal[1]:FREQuency:ALTErnate

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTErnate <val><units>
[:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTErnate?
```

This command sets the frequency for the alternate signal. The alternate signal frequency is the second tone of a dual-sine or the stop frequency of a swept-sine waveform.

Refer to [:AM\[1\]|2:INTernal\[1\]|2:FUNCTion:SHAPE](#) command for the waveform selection.

### Example

```
:AM2:INT1:FREQ:ALT 500KHZ
```

The preceding example sets the alternate frequency (AM path 2, AM internal source 1) for AM tone 2 to 500 kHz.

```
*RST                +4.00000000E+002
Range               Dual-Sine: 0.5 Hz to 1 MHz    Swept-Sine: 1 Hz to 1 MHz
Key Entry          AM Tone 2 Rate    AM Stop Rate
```

## **:AM[1]|2:INteRnal[1]:FREQuency:ALteRnate:AMPLitude:PERCent**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INteRnal[1]:FREQuency:ALteRnate:AMPLitude:  
PERCent <val> [ :SOURce]:AM[1]|2:INteRnal[1]:FREQuency:ALteRnate:AMPLitude:PERCent?
```

This command sets the amplitude of the second tone for a dual-sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude.

Refer to [:AM\[1\]|2:INteRnal\[1\]|2:FUNCTion:SHAPE](#) command for the waveform selection.

### **Example**

```
:AM2:INT1:FREQ:ALT:AMPL:PERC 50
```

The preceding example sets the amplitude (AM path 2, AM internal source 1) for AM tone 2 to 50% of the total amplitude.

**\*RST** +5.00000000E+001

**Range** 0 to 100 percent

**Key Entry** **AM Tone 2 Ampl Percent Of Peak**

## **:AM[1]|2:INteRnal[1]|2:FUNCTion:NOISe**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INteRnal[1]|2:FUNCTion:NOISe GAUSSian|UNIForm  
[ :SOURce]:AM[1]|2:INteRnal[1]|2:FUNCTion:NOISe?
```

This command selects a gaussian or uniform noise modulation for the selected waveform.

Refer to [:AM\[1\]|2:INteRnal\[1\]|2:FUNCTion:SHAPE](#) command for the waveform selection.

### **Example**

```
:AM2:INT1:FUNC:NOIS GAUS
```

The preceding example selects the gaussian noise waveform for AM modulation on AM path 2, internal source 1.

**\*RST** UNIF

**Key Entry** **Gaussian Uniform**

## **:AM[1]|2:INteRnal[1]|2:FUNCTion:RAMP**

**Supported** All models with Option 007 and UNT

```
[ :SOURce]:AM[1]|2:INteRnal[1]|2:FUNCTion:RAMP POSitive|NEGative  
[ :SOURce]:AM[1]|2:INteRnal[1]|2:FUNCTion:RAMP?
```

This command selects a positive or negative slope for the modulating ramp waveform.

Refer to [:AM\[1\]|2:INteRnal\[1\]|2:FUNCTion:SHAPE](#) command for the waveform selection.

### **Example**

```
:AM2:INT1:FUNC:RAMP NEG
```

The preceding example sets the slope of the ramp modulation for AM path 2, internal source 1, to negative.

```
*RST                POS
Key Entry          Positive   Negative
```

### **:AM[1]|2:INTernal[1]|2:FUNction:SHAPE**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INTernal[1]|2:FUNction:SHAPE SINE|TRIangle|SQUare|
RAMP|NOISe|DUALsine|SWEptsine
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNction:SHAPE?
```

This command sets the AM waveform type. The INTernal2 source selection does not support the dual-sine or Sweep-Sine waveform choices.

#### **Example**

```
:AM1:INT1:FUNC:SHAP DUAL
```

The preceding example sets the AM waveform type for AM path 1, internal source 1, to dual sine.

```
*RST                SINE
Key Entry          Sine     Triangle   Square    Ramp     Noise    Dual-Sine   Swept-Sine
```

### **:AM[1]|2:INTernal[1]:SWEep:RATE**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE <val><units>
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE?
```

This command sets the sweep rate for the AM swept-sine waveform.

For the waveform selection, refer to the [:AM\[1\]|2:INTernal\[1\]|2:FUNction:SHAPE](#) command. The sweep rate function is only available for internal source 1.

#### **Example**

```
:AM2:INT1:SWE:RATE 1KHZ
```

The preceding example sets the sweep rate for AM path 1 and internal source 1 to 1 kHz.

```
*RST                +4.00000000E+002
Range              0.5 Hz to 100 kHz
Key Entry          AM Sweep Rate
```

### **:AM[1]|2:INTernal[1]:SWEep:TRIGger**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:INTernal[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:TRIGger?
```

This command sets the trigger source for the AM swept-sine waveform.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN triggering using the *TRG command.
IMMEDIATE	This choice enables immediate triggering of the sweep event.
EXTERNAL	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables triggering through front panel interaction by pressing the <b>Trigger</b> hardkey.

For the waveform selection, refer to the :AM[1]|2:INTERNAL[1]|2:FUNCTION:SHAPE command.

**Example**

```
:AM1:INT1:SWE:TRIG EXT
```

The preceding example sets an external trigger source for the swept-sine waveform on AM path 1.

*RST	IMM			
<b>Key Entry</b>	<b>Bus</b>	<b>Free Run</b>	<b>Ext</b>	<b>Trigger Key</b>

**:AM[1]|2:SOURce**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2  

[:SOURce]:AM[1]|2:SOURce?
```

This command selects the source for amplitude modulation.

INT	This choice selects internal source 1 or 2 to provide an ac-coupled signal.
EXT	This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac- or dc-coupled. The externally applied, ac-coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front panel display, will indicate a high or low condition if that voltage is > ±3% of 1 V <sub>p</sub> .

**Example**

```
:AM2:SOUR INT1
```

The preceding example selects internal source 1 as the source for AM path 2.

*RST	INT			
<b>Key Entry</b>	<b>Internal 1</b>	<b>Internal 2</b>	<b>Ext1</b>	<b>Ext2</b>

**:AM[1]|2:STATe**

**Supported** All models with Option UNT

```
[ :SOURce]:AM[1]|2:STATe ON|OFF|1|0  

[:SOURce]:AM[1]|2:STATe?
```

This command enables or disables amplitude modulation for the selected path.

The RF carrier is modulated when the instrument's `:MODulation[:STATe]` command is enabled ON. Whenever amplitude modulation is enabled, the AM annunciator appears on the signal generator's front panel display.

The two paths can be simultaneously enabled (ON), by the `:AM[1]|2...` command.

**Example**

```
:AM1:STAT ON
```

The preceding example turns on AM modulation for AM path 1.

```
*RST          0
```

**Key Entry**           **AM Off On**

**:AM[1]|2:TYPE**

**Supported**           All models with Option UNT

```
[ :SOURce]:AM[1]|2:TYPE LINear|EXPonential
```

```
[ :SOURce]:AM[1]|2:TYPE?
```

This command sets the AM type to linear or exponential AM.

**LINear**               This choice selects linear AM type with depth values in units of percent/volt.

**EXPonential**         This choice selects exponential AM type with depth values in units of dB/volt.

**Example**

```
:AM2:TYPE EXP
```

The preceding example selects exponential type depth values for AM path 2.

```
*RST          LIN
```

**Key Entry**           **AM Type LIN EXP**

**:AM[1]|2[:DEPTh]:EXPonential**

**Supported**           All models with Option UNT

```
[ :SOURce]:AM[1]|2[:DEPTh]:EXPonential <val>
```

```
[ :SOURce]:AM[1]|2[:DEPTh]:EXPonential?
```

This commands sets the AM depth in dB/volt units. EXPonential must be the current AM type for this command to have any affect. Refer to the `:AM[1]|2:TYPE` command.

**Example**

```
:AM2:EXP 20
```

The preceding example sets the exponential depth to 20 dB for AM path 2.

```
*RST          +4.00000000E+001
```

**Range**               0.00 to 40.00 dB

**Key Entry**           **AM Depth**

## **:AM[1] | 2[:DEPTh][:LINear]**

**Supported** All models with Option UNT

```
[ :SOURce ] :AM[ 1 ] | 2 [ :DEPT h ] [ :LINear ] <val> | UP | DOWN  
[ :SOURce ] :AM[ 1 ] | 2 [ :DEPT h ] [ :LINear ] ?
```

This command sets the AM depth in percent/volt units. The command, used with the UP|DOWN parameters, will change the depth by a user-defined step value. The value associated with the UP and DOWN choices, are set by the **:AM[:DEPT h]:STEP[:INCRement]** command.

LINear must be the current AM type for this command to have any affect. To set the AM measurement type, use the **:AM[1]|2:TYPE** command. When the depth values are coupled, a change made to one path is applied to both. For AM depth value coupling, refer to the **:AM[1]|2[:DEPT h][:LINear]:TRACk** command.

### **Example**

```
:AM2 20
```

The preceding example sets the AM path 2 linear depth to 20%.

**\*RST** +1.00000000E-001

**Range** 0.0 to 100 percent

**Key Entry** **AM Depth**

## **:AM[1] | 2[:DEPT h][:LINear]:TRACk**

**Supported** All models with Option UNT

```
[ :SOURce ] :AM[ 1 ] | 2 [ :DEPT h ] [ :LINear ] :TRACk ON | OFF | 1 | 0  
[ :SOURce ] :AM[ 1 ] | 2 [ :DEPT h ] [ :LINear ] :TRACk ?
```

This command enables or disables AM depth value coupling between AM paths 1 and 2. When the depth values are coupled, a change made to one path is applied to both. LINear must be the AM type for this command to have any affect. To set the AM measurement type, use the **:AM[1]|2:TYPE** command.

ON (1) This choice will link the depth value of AM[1] with AM2; AM2 will assume the AM[1] depth value. For example, if AM[1] depth is set to 15% and AM2 is set to 11%, enabling the depth tracking will cause the AM2 depth value to change to 15%. This applies regardless of the path (AM[1] or AM2) selected in this command

OFF (0) This choice disables coupling and both paths will have independent depth values.

### **Example**

```
:AM1:TRAC ON
```

The preceding example enables AM depth coupling between AM path 1 and AM path 2.

**\*RST** 0

**Key Entry** **AM Depth Couple Off On**

## **:AM[:DEPTh]:STEP[:INCRement]**

**Supported** All models with Option UNT

```
[ :SOURCE ] :AM [ :DEPTh ] :STEP [ :INCRement ] <val> | MAXimum | MINimum | DEFault  
[ :SOURCE ] :AM [ :DEPTh ] :STEP [ :INCRement ] ?
```

This command sets the linear depth step value in percent/volt units.

The step value set by this command is used with the UP and DOWN choices for the [:AM\[1\]|2\[:DEPTh\]\[:LINear\]](#) command.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:AM:STEP 10
```

The preceding example sets the step value for AM depth to 10%.

**Range** 0.1 to 100 percent

**Key Entry** **Incr Set**

## Frequency Modulation Subsystem ([:SOURce])

**:FM[1]|2...**

**Supported**            All Models

[[:SOURce]:FM[1]|2...]

This prefix enables the selection of the FM path and is associated with all SCPI commands in this subsystem. The two paths are equivalent to the **FM Path 1 2** softkey.

FM1                    **FM Path 1 2** with 1 selected

FM2                    **FM Path 1 2** with 2 selected

When just FM is shown in a command, this means the command applies to path one only.

Each path is set up separately. When a SCPI command uses FM1, only path one is affected. Consequently, when FM2 is selected, only path two is set up. However, the deviation of the signals for the two paths can be coupled.

Deviation coupling links the deviation value of FM1 to FM2. Changing the deviation value for one path changes it for the other. These two paths can be on at the same time provided the following conditions have been met:

- dual-sine or swept-sine is not the selection for the waveform type
- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)
- FM2 must be set to a deviation less than FM1



### **:FM:INTernal:FREQuency:STEP[:INCRement]**

**Supported** All models with Option UNT

```
[ :SOURCE]:FM:INTernal:FREQuency:STEP[ :INCRement ] <num> | MAXimum | MINimum | DEFault
[:SOURCE]:FM:INTernal:FREQuency:STEP[ :INCRement ] ?
```

This command sets the step value for the internal frequency modulation.

The step value set by this command is used with the UP and DOWN choices for the [:FM\[1\]|2:INTernal\[1\]|2:FREQuency](#) command.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

#### **Example**

```
:FM:INT:FREQ:STEP 1E5
```

The preceding example sets the step value to .1 MHz.

**Range** 0.5 to 1E6

### **:FM[1]|2:EXTernal[1]|2:COUPLing**

**Supported** All models with Option UNT

```
[ :SOURCE]:FM[1]|2:EXTernal[1]|2:COUPLing AC|DC
[:SOURCE]:FM[1]|2:EXTernal[1]|2:COUPLing ?
```

This command sets the coupling type for the selected external input. The command does not change the active source or switch modulation on or off. The modulating signal may be the sum of several signals, from either internal or external sources.

AC This choice will pass only ac signal components.

DC This choice will pass both ac and dc signal components.

#### **Example**

```
:FM1:EXT1:COUP AC
```

The preceding example sets the coupling for FM path 1, external source 1 to AC.

\*RST DC

**Key Entry** Ext Coupling DC AC

### **:FM[1]|2:EXTernal[1]|2:IMPedance**

**Supported** All models with Option UNT

```
[ :SOURCE]:FM[1]|2:EXTernal[1]|2:IMPedance <50|600>
[:SOURCE]:FM[1]|2:EXTernal[1]|2:IMPedance ?
```

This command sets the impedance for the external input.

### Example

```
:FM1:EXT2:IMP 600
```

The preceding example sets the FM path 1, external 1 source impedance to 600 ohms.

```
*RST +5.00000000E+001
```

**Key Entry**            **Ext Impedance 50 Ohm 600 Ohm**

### :FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE

**Supported**            All models with Option UNT

```
[ :SOURce]:FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE <val><units>
```

```
[ :SOURce]:FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE?
```

This command sets the internal FM rate of the alternate signal. The alternate signal frequency is the second tone of a dual-sine or the stop frequency of a swept-sine waveform.

For waveform selection, use the [:FM\[1\]|2:INTERNAL\[1\]|2:FUNCTION:SHAPE](#) command.

### Example

```
:FM1:INT:FREQ:ALT 20KHZ
```

The preceding example sets the FM tone 2 rate for FM path 1, FM source 1, to 20 kHz.

```
*RST +4.00000000E+002
```

**Range**                *dual-sine*: 0.5 Hz to 100 kHz      *swept-sine*: 0.5 Hz to 100 kHz

**Key Entry**            **FM Tone 2 Rate**    **FM Stop Rate**

### :FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE:AMPLITUDE:PERCENT

**Supported**            All models with Option UNT

```
[ :SOURce]:FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE:AMPLITUDE:
```

```
PERCENT <val><units>
```

```
[ :SOURce]:FM[1]|2:INTERNAL[1]:FREQUENCY:ALTERNATE:AMPLITUDE:PERCENT?
```

This command sets the amplitude of the second tone for a dual-sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude.

For waveform selection, use the [:FM\[1\]|2:INTERNAL\[1\]|2:FUNCTION:SHAPE](#) command.

### Example

```
:FM1:INT:FREQ:ALT:AMPL:PERC 20
```

The preceding example sets the amplitude for FM tone 2, FM path 1, FM internal source 1 to 20% of the total amplitude.

```
*RST +5.00000000E+001
```

**Range**                0 to 100 percent

**Key Entry**            **FM Tone 2 Ampl Percent Of Peak**

## **:FM[1]|2:INteRnal[1]:SWEep:RATE**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]:SWEep:RATE <val><units>
[:SOURce]:FM[1]|2:INteRnal[1]:SWEep:RATE?
```

This command sets the sweep rate for the swept-sine waveform. The minimum resolution is 0.5 hertz. For waveform selection, use the **:FM[1]|2:INteRnal[1]|2:FUNcTION:SHApe** command.

### **Example**

```
:FM1:INT:SWE:RATE 20KHZ
```

The preceding example sets the sweep rate for the swept-sine waveform to 20 kilohertz.

**\*RST** +4.00000000E+002

**Range** 0.5 Hz to 100 kHz

**Key Entry** **FM Sweep Rate**

## **:FM[1]|2:INteRnal[1]:SWEep:TRIGger**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]:SWEep:TRIGger BUS|IMMediate|EXteRnal|KEY
[:SOURce]:FM[1]|2:INteRnal[1]:SWEep:TRIGger?
```

This command sets the trigger source for the FM swept-sine waveform. For waveform selection, use the **:FM[1]|2:INteRnal[1]|2:FUNcTION:SHApe** command.

**BUS** This choice enables GPIB triggering using the \*TRG or GET command or LAN triggering using the \*TRG command.

**IMMediate** This choice enables immediate triggering of the sweep event. This choice is equivalent to pressing the **Free Run** softkey.

**EXteRnal** This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.

**KEY** Enables triggering through front panel interaction (the **Trigger** hardkey).

**\*RST** IMM

### **Example**

```
:FM1:INT:SWE:TRIG BUS
```

The preceding example selects the bus as the trigger source for FM path 1.

**Key Entry** **Bus** **Free Run** **Ext** **Trigger Key**

## **:FM[1]|2:INteRnal[1]|2:FREQuency**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]|2:FREQuency <val><units>|UP|DOWN
[:SOURce]:FM[1]|2:INteRnal[1]|2:FREQuency?
```

This command sets the internal FM rate using the <val><units> variable, or changes the FM rate by a user-defined up/down step value. Refer to the [:FM:INteRnal:FREQuency:STEP\[:INCRement\]](#) command for setting the value associated with the UP and DOWN choices.

The command changes:

- the FM rate of the first tone of a dual-sine waveform
- the starting FM rate for a swept-sine waveform
- the FM rate for all other waveforms

For selecting the waveform, use the [:FM\[1\]|2:INteRnal\[1\]|2:FUNcTION:SHAPE](#) command.

### **Example**

```
:FM2:INT:FREQ 40KHZ
```

The preceding example sets the modulation rate for FM path 2 to 40 kHz.

```
*RST +4.00000000E+002
```

**Range** *Dual-Sine & Sine:* 0.5 Hz to 1 MHz *Swept-Sine:* 1 Hz to 1MHz  
*All Other Waveforms:* 0.5 Hz to 100 kHz

<b>Key Entry</b>	<b>FM Tone 1 Rate</b>	<b>FM Start Rate</b>	<b>FM Rate</b>
------------------	-----------------------	----------------------	----------------

## **:FM[1]|2:INteRnal[1]|2:FUNcTION:NOISe**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTION:NOISe GAUSSian|UNIForm
[:SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTION:NOISe?
```

This command selects a gaussian or uniform noise type as the modulation. For waveform selection, use the [:FM\[1\]|2:INteRnal\[1\]|2:FUNcTION:SHAPE](#) command.

### **Example**

```
:FM2:INT2:FUNC:NOIS UNIF
```

The preceding example selects a uniform noise waveform as the modulation for FM path 2 and FM source 2.

```
*RST UNIF
```

<b>Key Entry</b>	<b>Gaussian</b>	<b>Uniform</b>
------------------	-----------------	----------------

## **:FM[1]|2:INteRnal[1]|2:FUNcTion:RAMP**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTion:RAMP POSitive|NEGative
[:SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTion:RAMP?
```

This command selects a positive or negative ramp as the internal modulating waveform. Refer to [:FM\[1\]|2:INteRnal\[1\]|2:FUNcTion:SHAPE](#) command for the waveform selection.

### **Example**

```
:FM2:INT2:FUNC:RAMP POS
```

The preceding example selects a positive sloped ramp as the internal modulating waveform.

```
*RST POS
```

**Key Entry**            **Positive**    **Negative**

## **:FM[1]|2:INteRnal[1]|2:FUNcTion:SHAPE**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTion:SHAPE SINE|TRIangle|SQUare|
RAMP|NOISe|DUALsine|SWEPTsine
[:SOURce]:FM[1]|2:INteRnal[1]|2:FUNcTion:SHAPE?
```

This command selects the FM waveform type. The INteRnal2 source selection does not support the dual-sine or Sweep-Sine waveform types.

### **Example**

```
:FM2:INT1:FUNC:SHAP SQU
```

The preceding example selects a square wave as the internal modulating waveform.

```
*RST SINE
```

**Key Entry**            **Sine**    **Triangle**    **Square**    **Ramp**    **Noise**    **Dual-Sine**    **Swept-Sine**

## **:FM[1]|2:SOURce**

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:SOURce INT[1]|INT2|EXT1|EXT2
[:SOURce]:FM[1]|2:SOURce?
```

This command selects the FM source.

**INT** This choice selects internal source 1 or 2 to provide an ac-coupled signal.

**EXT** This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac- or dc-coupled. The externally applied, ac-coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front panel display, will indicate a high or low condition if that voltage is  $> \pm 3\%$  of  $1 V_p$ .

### Example

```
:FM2:SOUR INT2
```

The preceding example selects internal source 2 as the FM source for FM path 2.

```
*RST          INT
Key Entry     Internal 1   Internal 2   Ext1   Ext2
```

### :FM[1]|2:STATe

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2:STATe ON|OFF|1|0
[:SOURce]:FM[1]|2:STATe?
```

This command enables or disables the selected FM path.

The RF carrier is modulated when the instrument's **:MODulation[:STATe]** command is enabled ON.

Whenever frequency modulation is enabled, the FM annunciator appears on the signal generator's front panel display.

The two paths for frequency modulation can be simultaneously enabled ON, with the **:FM[1]|2...** command.

### Example

```
:FM2:STAT ON
```

The preceding example enables FM path 2.

```
*RST          0
Key Entry     FM Off On
```

### :FM[1]|2[:DEVIation]

**Supported** All models with Option UNT

```
[ :SOURce]:FM[1]|2[:DEVIation] <val><units>
[:SOURce]:FM[1]|2[:DEVIation]?
```

This command sets the FM deviation for the selected FM path.

If deviation tracking is ON, a change to the deviation value on one path will apply to both. To set the deviation tracking, use the **:FM[1]|2[:DEVIation]:TRACk** command.

### Example

```
:FM2 1MHZ
```

The preceding example sets the frequency deviation to 1 megahertz.

<b>*RST</b>	+1.00000000E+003	
<b>Range</b>	<b>Frequency</b>	<b>Deviation</b>
	250KHZ–250MHZ	0–2MHz
	> 250–500MHZ	0–1MHz
	> 0.5–1GHZ	0–2MHz
	> 1–2GHZ	0–4MHz
	> 2–3.2GHZ	0–8MHz
	> 3.2–10GHZ	0–16MHz
	> 10–20GHZ	0–32MHz
	> 20–28.5GHZ <sup>a</sup>	0–48MHz
	> 20–40GHZ	0–64MHz
	> 28.5–44GHZ <sup>a</sup>	0–80MHz
	> 40–67GHZ	0–128MHz

a. E8267D Only

**Key Entry**            **FM DEV**

**:FM[1]|2[:DEVIation]:TRACk**

**Supported**            All models with Option UNT

[ :SOURce]:FM[1]|2[:DEVIation]:TRACk ON|OFF|1|0

[ :SOURce]:FM[1]|2[:DEVIation]:TRACk?

This command enables or disables deviation coupling between FM paths 1 and 2.

- ON (1)                    This choice will link the deviation value of FM1 with FM2; FM2 will assume the FM1 deviation value. For example, if FM1 deviation is set to 500 Hz and FM2 is set to 2 kHz, enabling the deviation tracking will cause the FM2 deviation value to change to 500 Hz. This applies regardless of the path (FM1 or FM2) selected.
- OFF (0)                   This choice disables the coupling and both paths will have independent deviation values.

This command uses exact match tracking, not offset tracking.

**Example**

:FM2:TRAC 0

The preceding example disables deviation coupling.

**\*RST**                    0

**Key Entry**            **FM Dev Couple Off On**

## Low Frequency Output Subsystem ([:SOURce]:LFOutput)

### :LFOutput:AMPLitude

**Supported** All models with Option UNT

```
[ :SOURce ] :LFOutput :AMPLitude <val> <units>
```

```
[ :SOURce ] :LFOutput :AMPLitude ?
```

This command sets the amplitude of the signal at the LF OUTPUT connector.

#### Example

```
:LFO:AMPL 2.1VP
```

The preceding example sets the peak amplitude to 2.1 volts.

**\*RST** 0.00

**Range** 0.000 to 3.5 VP

**Key Entry** LF Out Amplitude

### :LFOutput:FUNCTION[1] | 2:FREQuency

**Supported** All models with Option UNT

```
[ :SOURce ] :LFOutput :FUNCTION[1] | 2:FREQuency <val> <units>
```

```
[ :SOURce ] :LFOutput :FUNCTION[1] | 2:FREQuency ?
```

This command sets the frequency of function generator 1 or 2. The command sets:

- the frequency of the first tone of a dual-sine waveform
- the start frequency for a swept-sine waveform
- the frequency for all other waveform types

For selecting the waveform type, use the [:LFOutput:FUNCTION\[1\] | 2:SHAPE](#) command.

#### Example

```
:LFO:FUNC1:FREQ .1MHZ
```

The preceding example sets the frequency for function generator 1 to 100 kHz.

**\*RST** +4.00000000E+002

**Range** *Sine and Dual-Sine:* 0.5 Hz to 1 MHz

**Range** *Swept-Sine:* 1 Hz to 1 MHz

*All Other Waveforms:* 0.5 Hz to 100 kHz

**Key Entry** LF Out Tone 1 Freq LF Out Start Freq LF Out Freq



## **:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE**

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE <val><units>
[:SOURce]:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE?
```

This command sets the frequency for the alternate LF output signal. The alternate frequency is the second tone of a dual-sine or the stop frequency of a swept-sine waveform.

For selecting the waveform type, use the [:LFOOutput:FUNCTION\[1\]:2:SHAPE](#) command.

### **Example**

```
:LFO:FUNCL:FREQ:ALT 20KHZ
```

The preceding example sets the alternate frequency to 20 kHz.

**\*RST** +4.00000000E+002

**Range** *Dual-Sine*: 0.1 Hz to 100 kHz *Swept-Sine*: 0.1 Hz to 100 kHz

**Key Entry** **LF Out Tone 2 Freq** **LF Out Stop Freq**

## **:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE:AMPLITUDE:PERCENT**

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE:AMPLITUDE:
PERCENT <val><units>
[:SOURce]:LFOOutput:FUNCTION[1]:FREQUENCY:ALTERNATE:AMPLITUDE:PERCENT?
```

This command sets the amplitude of the second tone for a dual-sine waveform as a percentage of the total LF output amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude.

For selecting the waveform type, use the [:LFOOutput:FUNCTION\[1\]:2:SHAPE](#) command.

### **Example**

```
:LFO:FUNCL:FREQ:ALT:AMPL:PERC 50
```

The preceding example sets the alternate frequency to 50% of the total output amplitude.

**\*RST** +5.00000000E+001

**Range** 0 to 100 percent

**Key Entry** **LF Out Tone 2 Ampl % of Peak**

## **:LFOOutput:FUNCTION[1]:2:SHAPE**

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:FUNCTION[1]:2:SHAPE SINE|DUALsine|SWEPTSine|TRIangle|
SQUare|RAMP|PULSe|NOISE|DC
[:SOURce]:LFOOutput:FUNCTION[1]:2:SHAPE?
```

This command selects the waveform type. Function Generator 1 must be the source for the dual-sine or the swept-sine waveform.

Refer to “:LFOutput:SOURce” on page 180.

### Example

```
:LFO:FUNC2:SHAP TRI
```

The preceding example selects a triangle wave for the Function Generator 2 LF output.

```
*RST SINE
```

<b>Key Entry</b>	<b>Sine</b>	<b>Dual-Sine</b>	<b>Swept-Sine</b>	<b>Triangle</b>	<b>Square</b>	<b>Ramp</b>	<b>Pulse</b>
	<b>Noise</b>	<b>DC</b>					

### :LFOutput:FUNCTION:[1]|2:SHAPE:NOISE

**Supported** All models with Option UNT

```
[ :SOURce]:LFOutput:FUNCTION[1]|2:SHAPE:NOISE UNIFORM|GAUSSIAN  
[:SOURce]:LFOutput:FUNCTION[1]|2:SHAPE:NOISE?
```

This command selects a gaussian or uniform noise modulation for the LF output.

For selecting the waveform type, use the :LFOutput:FUNCTION[1]|2:SHAPE command.

### Example

```
:LFO:FUNC1:SHAP:NOIS GAUS
```

The preceding example selects a gaussian noise modulation for the Function Generator 1 LF output.

```
*RST UNIF
```

<b>Key Entry</b>	<b>Uniform</b>	<b>Gaussian</b>
------------------	----------------	-----------------

### :LFOutput:FUNCTION[1]|2:SHAPE:RAMP

**Supported** All models with Option UNT

```
[ :SOURce]:LFOutput:FUNCTION[1]|2SHAPE:RAMP POSITIVE|NEGATIVE  
[:SOURce]:LFOutput:FUNCTION[1]|2SHAPE:RAMP?
```

This command selects a positive or negative slope for the ramp modulation on the LF output.

For selecting the waveform type, use the :LFOutput:FUNCTION[1]|2:SHAPE command.

### Example

```
:LFO:FUNC1:SHAP:RAMP POS
```

The preceding example selects a positive ramp slope modulation for the Function Generator 1 LF output.

```
*RST POS
```

<b>Key Entry</b>	<b>Positive</b>	<b>Negative</b>
------------------	-----------------	-----------------

## **:LFOOutput:FUNCTION[1]:SWEep:RATE**

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:FUNCTION[1]:SWEep:RATE <val><units>
[:SOURce]:LFOOutput:FUNCTION[1]:SWEep:RATE?
```

This command sets the sweep rate for an internally generated swept-sine signal.

### **Example**

```
:LFO:FUNC1:SWE:RATE 1E5
```

The preceding example sets the sweep rate for the swept-sine waveform to 100 kHz.

**\*RST** +4.00000000E+002

**Range** 0.5 Hz to 100 kHz

**Key Entry** **LF Out Sweep Rate**

## **:FUNCTION[1]:SWEep:TRIGger**

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:FUNCTION[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY
[:SOURce]:LFOOutput:FUNCTION[1]:SWEep:TRIGger?
```

This command sets the trigger source for the internally generated swept-sine signal at the LF output.

**BUS** This choice enables GPIB triggering using the \*TRG or GET command or LAN and RS-232 triggering using the \*TRG command.

**IMMediate** This choice enables immediate triggering of the sweep event.

**EXTernal** This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.

**KEY** This choice enables triggering through front panel interaction by pressing the **Trigger** hardkey.

For selecting the waveform type, use the **:LFOOutput:FUNCTION[1]|2:SHAPE** command.

### **Example**

```
:LFO:FUNC1:SWE:TRIG EXT
```

The preceding example sets an external trigger as the trigger for the swept-sine signal.

**\*RST** Free Run

**Key Entry** **Bus Free Run Ext Trigger Key**

## :LFOOutput:SOURce

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:SOURce INT[1]|INT2|FUNCTION[1]|FUNCTION2  
[:SOURce]:LFOOutput:SOURce?
```

This command selects the source for the LF output.

**INT** This choice enables you to output a signal where the frequency and shape of the signal is set by internal source 1 or 2. For example, if the internal source is currently assigned to an AM path configuration and AM is turned on, the signal output at the LF OUTPUT connector will have the frequency and shape of the amplitude modulating signal.

**FUNCTION** This choice enables the selection of an internal function generator.

### Example

```
:LFO:SOUR FUNC1
```

The preceding example selects Function Generator 1 as the active LF source.

**\*RST** INT

<b>Key Entry</b>	<b>Internal 1 Monitor Function Generator 1</b>	<b>Internal 2 Monitor Function Generator 2</b>
------------------	--	--

## :LFOOutput:STATe

**Supported** All models with Option UNT

```
[ :SOURce]:LFOOutput:STATe ON|OFF|1|0  
[:SOURce]:LFOOutput:STATe?
```

This command enables or disables the low frequency output.

### Example

```
:LFO:STAT ON
```

The preceding example enables the source.

**\*RST** 0

<b>Key Entry</b>	<b>LF Out Off On</b>
------------------	----------------------

## Phase Modulation Subsystem ([:SOURce])

**:PM[1]|2...**

**Supported** All Models

[ :SOURce ] :PM[ 1 ] | 2 . . .

This prefix enables the selection of the  $\Phi$ M path and associated with all SCPI commands in this subsystem. The two paths are equivalent to the  $\Phi$ M Path 1 2 softkey.

PM1  $\Phi$ M Path 1 2 with 1 selected

PM2  $\Phi$ M Path 1 2 with 2 selected

When just PM is shown in a command, this means the command applies to path 1 only.

Each path is set up separately. When a SCPI command uses PM1, only path one is affected. Consequently, when PM2 is selected, only path two is set up. However, the deviation of the signals for the two paths can be coupled.

Deviation coupling links the deviation value of PM1 to PM2. Changing the deviation value for one path will change it for the other path. These two paths can be on at the same time provided the following conditions have been met:

- dual-sine or Sweep-Sine is not the selection for the waveform type
- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)
- PM2 must be set to a deviation less than or equal to PM1

**:PM:INTernal:FREQuency:STEP[:INCRement]**

**Supported** All models with Option UNT

[ :SOURce ] :PM:INTernal:FREQuency:STEP[ :INCRement ] <num> | MAXimum | MINimum | DEFault  
[ :SOURce ] :PM:INTernal:FREQuency:STEP[ :INCRement ] ?

This command sets the step value of the phase modulation internal frequency.

The step value set by this command is used with the UP and DOWN choices for the [:PM\[1\]|2:INTernal\[1\]:FREQuency](#) command.

The setting enabled by this command is not affected by a signal generator power-on, preset, or \*RST command.

### Example

```
:PM:INT:FREQ:STEP 1E5
```

The preceding example sets the step value to 100 kHz.

**Range** 0.5 to 1E6

**Key Entry** **Incr Set**

### **:PM[1]|2:BANDwidth|BWIDth**

**Supported** All models with Option UNT

[[:SOURCE]:PM[1]|2:BANDwidth|BWIDth NORMAL|HIGH

[[:SOURCE]:PM[1]|2:BANDwidth|BWIDth?

This command selects normal phase modulation or high bandwidth phase modulation. The command can use either the BANDwidth or BWIDth syntax.

#### **Example**

```
:PM1:BAND NORM
```

The preceding example selects normal phase modulation for  $\Phi$ M path 1.

**\*RST** NORM

**Key Entry** FM  $\Phi$ M Normal High BW

### **:PM[1]|2:EXTErnal[1]|2:COUPling**

**Supported** All models with Option UNT

[[:SOURCE]:PM[1]|2:EXTErnal[1]|2:COUPling AC|DC

[[:SOURCE]:PM[1]|2:EXTErnal[1]|2:COUPling?

This command sets the coupling for the phase modulation source at the selected external input connector.

AC This choice will only pass ac signal components.

DC This choice will pass both ac and dc signal components.

This command does not change the active source or switch modulation on or off. The modulating signal may be the sum of several signals, from either internal or external sources.

#### **Example**

```
:PM1:EXT:COUP AC
```

The preceding example selects AC coupling at the external input for  $\Phi$ M path 1.

**\*RST** DC

**Key Entry** Ext Coupling DC AC

### **:PM[1]|2:EXternal[1]|2:IMPedance**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2:EXternal[1]|2:IMPedance <50|600>
[:SOURce]:PM[1]|2:EXternal[1]|2:IMPedance?
```

This command selects 50 ohms or 600 ohms as the input impedance for the external input signal.

#### **Example**

```
:PM1:EXT2:IMP 600
```

The preceding example sets the  $\Phi$ M path 1, external 2 source impedance to 600 ohms.

```
*RST +5.00000000E+001
```

**Key Entry** Ext Impedance 50 Ohm 600 Ohm

### **:PM[1]|2:INTERNAL[1]:FREQUENCY**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2:INTERNAL[1]|2:FREQUENCY <val><units>
[:SOURce]:PM[1]|2:INTERNAL[1]|2:FREQUENCY?
```

This command sets the internal modulation frequency rate. The command sets:

- the frequency of the first tone of a dual-sine waveform
- the start frequency for a swept-sine waveform
- the frequency rate for all other waveforms

For selecting the waveform type, use the **:LFOutput:FUNCTION[1]|2:SHAPE** command.

#### **Example**

```
:PM1:INT1:FREQ 20KHZ
```

The preceding example sets the  $\Phi$ M path 1, internal source 1 frequency to 20 kHz.

```
*RST +4.00000000E+002
```

**Range** *Dual-Sine:* 0.1 Hz to 100 kHz *Swept-Sine:* 0.1 Hz to 100 kHz  
*All Other Waveforms:* 0.1 Hz to 20 kHz

**Key Entry**  $\Phi$ MTone 1 Rate  $\Phi$ M Start Rate  $\Phi$ M Rate

### **:PM[1]|2:INTERNAL[1]:FREQUENCY:ALternate**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2:INTERNAL[1]:FREQUENCY:ALternate <val><units>
[:SOURce]:PM[1]|2:INTERNAL[1]:FREQUENCY:ALternate?
```

This command sets the frequency rate for the alternate signal. The alternate frequency is the second tone of a dual-sine or the stop frequency of a swept-sine waveform.

To select the waveform, use the **:PM[1]|2:INTERNAL[1]:FUNCTION:SHAPE** command.

### Example

```
:PM1:INT1:FREQ:ALT 50KHZ
```

The preceding example sets the alternate frequency rate for the  $\Phi$ M tone 2,  $\Phi$ M path 1, source 1 to 50 kHz.

**\*RST** +4.00000000E+002

**Range** *Dual-Sine*: 0.1 Hz to 100 kHz *Swept-Sine*: 0.1 Hz to 100 kHz

**Key Entry**  **$\Phi$ M Stop Rate**  **$\Phi$ M Tone 2 Rate**

### :PM[1]|2:INTernal[1]|2:FUNCTION:NOISE

**Supported** All models with Option UNT

```
[ :SOURce ] : PM [ 1 ] | 2 : INTernal [ 1 ] | 2 : FUNCTION : NOISE GAUSSian | UNIFORM
```

```
[ :SOURce ] : PM [ 1 ] | 2 : INTernal [ 1 ] | 2 : FUNCTION : NOISE ?
```

This command selects a gaussian or uniform noise modulation for the selected path(s).

### Example

```
:PM1:INT1:FUNC:NOIS GAUS
```

The preceding example selects a gaussian noise modulation for  $\Phi$ M path 1, source 1.

**\*RST** UNIF

**Key Entry** **Gaussian** **Uniform**

### :PM[1]|2:INTernal[1]|2:FUNCTION:RAMP

**Supported** All models with Option UNT

```
[ :SOURce ] : PM [ 1 ] | 2 : INTernal [ 1 ] | 2 : FUNCTION : RAMP POSitive | NEGative
```

```
[ :SOURce ] : PM [ 1 ] | 2 : INTernal [ 1 ] | 2 : FUNCTION : RAMP ?
```

This command selects a positive or negative slope for the ramp modulating waveform.

### Example

```
:PM1:INT2:FUNC:RAMP POS
```

The preceding example selects a positive ramp slope for modulating the signal on  $\Phi$ M path 1, internal source 2.

**\*RST** POS

**Key Entry** **Positive** **Negative**



## **:PM[1]|2:INTernal[1]:FREQuency:ALTErnate:AMPLitude:PERCent**

**Supported** All models with Option UNT

```
[ :SOURCE]:PM[1]|2:INTernal[1]:FREQuency:ALTErnate:AMPLitude:
PERCent <val>
[:SOURCE]:PM[1]|2:INTernal[1]:FREQuency:ALTErnate:AMPLitude:PERCent?
```

This command sets the amplitude of the second tone for the dual-sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude.

To select the waveform, use the **:PM[1]|2:INTernal[1]:FUNCTion:SHAPE** command.

### **Example**

```
:PM2:INT:FREQ:ALT:AMPL:PERC 40
```

The preceding example sets the alternate tone amplitude to 40% of the total amplitude.

**\*RST** +5.00000000E+001

**Range** 0 to 100 percent

**Key Entry**  $\Phi$ M Tone 2 Ampl Percent of Peak

## **:PM[1]|2:INTernal[1]:FUNCTion:SHAPE**

**Supported** All models with Option UNT

```
[ :SOURCE]:PM[1]|2:INTernal[1]:FUNCTion:SHAPE SINE|TRIangle|SQUare|RAMP|
NOISE|DUALsine|SWEPTsine
[:SOURCE]:PM[1]|2:INTernal[1]:FUNCTion:SHAPE?
```

This command sets the phase modulation waveform type for internal source 1.

### **Example**

```
:PM1:INT:FUNC:SHAP RAMP
```

The preceding example selects a ramp modulation for  $\Phi$ M path 1, source 1.

**\*RST** SINE

**Key Entry** Sine Triangle Square Ramp Noise Dual-Sine Swept-Sine

## :PM[1]|2:INteRnal2:FUNction:SHApe

Supported All models with Option UNT

```
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal2 : FUNction : SHApe SINE | TRIangle | SQUare | RAMP | NOISe  
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal2 : FUNction : SHApe ?
```

This command sets the phase modulation waveform type for internal source 2.

### Example

```
:PM1:INT2:FUNC:SHAP RAMP
```

The preceding example selects a ramp modulation for  $\Phi$ M path 1, source 2.

**\*RST** SINE

**Key Entry** Sine Triangle Square Ramp Noise

## :PM[1]|2:INteRnal[1]:SWEep:RATE

Supported All models with Option UNT

```
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal [ 1 ] : SWEep : RATE <val><units>  
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal [ 1 ] : SWEep : RATE ?
```

This command sets the sweep rate for a phase-modulated, swept-sine waveform.

To select the waveform, use the [:PM\[1\]|2:INteRnal\[1\]:FUNction:SHApe](#) command.

### Example

```
:PM1:INT:SWE:RATE 30KHZ
```

The preceding example sets the sweep rate to 30 kHz.

**\*RST** +4.00000000E+002

**Range** 0.5 Hz to 100 kHz

**Key Entry**  $\Phi$ M Sweep Rate

## :PM[1]|2:INteRnal[1]:SWEep:TRIGger

Supported All models with Option UNT

```
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal [ 1 ] : SWEep : TRIGger BUS | IMMEDIATE | EXteRnal | KEY  
[ :SOURCE ] : PM [ 1 ] | 2 : INteRnal [ 1 ] : SWEep : TRIGger ?
```

This command sets the trigger source for the phase-modulated, swept-sine waveform.

**BUS** This choice enables GPIB triggering using the \*TRG or GET command or LAN and RS-232 triggering using the \*TRG command.

**IMMEDIATE** This choice enables immediate triggering of the sweep event. This choice is equivalent to pressing the **Free Run** softkey.

**EXteRnal** This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.

**KEY** This choice enables triggering through front panel interaction by pressing the **Trigger** hardkey.

To select the waveform, use the `:PM[1]|2:INTernal[1]:FUNCTion:SHAPE` command.

**Example**

```
:PM2:INT:SWE:TRIG BUS
```

The preceding example selects a BUS trigger as the triggering for the internal source 1 swept-sine waveform on  $\Phi$ M path 2.

```
*RST IMM
```

<b>Key Entry</b>	<b>Bus</b>	<b>Free Run</b>	<b>Ext</b>	<b>Trigger Key</b>
------------------	------------	-----------------	------------	--------------------

**:PM[1]|2:SOURce**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2  

[:SOURce]:PM[1]|2:SOURce?
```

This command selects the source used to generate the phase modulation.

**INT** This choice selects internal source 1 or internal source 2 to provide an ac-coupled signal.

**EXT** This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac- or dc-coupled.

The externally applied, ac-coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front panel display, will indicate a high or low condition if that voltage is  $> \pm 3\%$  of  $1 V_p$ .

**Example**

```
:PM2:SOUR EXT1
```

The preceding example selects an external signal on the EXT 1 INPUT connector as the source for  $\Phi$ M path 2 modulation.

```
*RST INT
```

<b>Key Entry</b>	<b>Internal 1</b>	<b>Internal 2</b>	<b>Ext1</b>	<b>Ext2</b>
------------------	-------------------	-------------------	-------------	-------------

**:PM[1]|2:STATe**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2:STATe ON|OFF|1|0  

[:SOURce]:PM[1]|2:STATe?
```

This command enables or disables the phase modulation for the selected path. The RF carrier is modulated when the instrument's `:MODulation[:STATe]` command is enabled ON.

The  $\Phi$ M annunciator appears on the signal generator's front panel display whenever phase modulation is enabled. The two paths for phase modulation can be simultaneously enabled ON by the `:PM[1]|2...` command.

### Example

```
:PM2:STAT 1
```

The preceding example turns on  $\Phi$ M path 2 phase modulation.

```
*RST 0
```

**Key Entry**  $\Phi$ M Off On

### :PM[1] | 2[:DEVIation]

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1] | 2[:DEVIation] <val><units> |UP|DOWN  

[:SOURce]:PM[1] | 2[:DEVIation]?
```

This command sets the deviation of the phase modulation. The variable <units> will accept RAD (radians), PIRAD (pi-radians), and DEG (degrees); however, the query will only return values in radians. If deviation tracking is active, a change to the deviation value on one path will apply to both.

The command, used with the UP|DOWN parameters, will change the deviation by a user-defined step value. For setting the value associated with the UP and DOWN choices, refer to the [:PM\[:DEVIation\]:STEP\[:INCRement\]](#) command.

### Example

```
:PM1 135DEG
```

The preceding example sets the phase modulation to 135 degrees.

```
*RST +0.00000000E+000
```

Range	Frequency	Normal Bandwidth	High Bandwidth
	250KHZ-250MHz	0-20rad	0-2rad
	> 250-500MHz	0-10rad	0-1rad
	> 0.5-1GHZ	0-20rad	0-2rad
	> 1-2GHZ	0-40rad	0-4rad
	> 2-3.2GHZ	0-80rad	0-8rad
	> 3.2-10GHZ	0-160rad	0-16rad
	> 10.0-20GHZ	0-320rad	0-32rad
	> 20.0-28.5GHZ <sup>a</sup>	0-480rad	0-48rad
	> 20.0-40.0GHZ	0-640rad	0-64rad
	> 28.5-44.0GHZ <sup>a</sup>	0-800rad	0-80rad
	>40-67.0GHZ <sup>b</sup>	0-1280rad	0-128rad

**Key Entry**  $\Phi$ M Dev

a. E8267D Only

b. Performance is not specified above 50 GHz

## **:PM[1]|2[:DEVIation]:TRACk**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[1]|2[:DEVIation]:TRACk ON|OFF|1|0
[:SOURce]:PM[1]|2[:DEVIation]:TRACk?
```

This command enables or disables the deviation coupling between the PM paths 1 and 2.

ON (1) This choice will link the deviation value of PM1 with PM2; PM2 will assume the PM[1] deviation value. For example, if PM1 deviation is set to 500 Hz and PM2 is set to 2 kHz, enabling the deviation tracking will cause the PM2 deviation value to change to 500 Hz. This applies regardless of the path (PM1 or PM2) selected in this command.

OFF (0) This choice disables the coupling and both paths will have independent deviation values.

This command uses exact match tracking, not offset tracking.

### **Example**

```
:PM1:TRAC OFF
```

The preceding example disables deviation coupling.

```
*RST 0
```

**Key Entry**  $\Phi$ M Dev Couple Off On

## **:PM[:DEVIation]:STEP[:INCRement]**

**Supported** All models with Option UNT

```
[ :SOURce]:PM[:DEVIation]:STEP[:INCRement]<val><units>|MAXimum|MINimum|DEFault
[:SOURce]:PM[:DEVIation]:STEP[:INCRement]?
```

This command sets the phase modulation deviation step value.

The value set by this command is used with the UP and DOWN choices for the FM deviation command.

The setting is not affected by a signal generator power-on, preset, or \*RST command.

See also: “:PM[1]|2[:DEVIation]” on page 188.

### **Example**

```
:PM:STEP 20RAD
```

The preceding example sets the step value to 20 radians.

**Range** 0.001 to 1E3 radians

## Pulse Modulation Subsystem ([:SOURce])

### :PULM:EXTernal:POLarity NORMAL|INVerted

**Supported** All models with Option UNU or UNW

```
[ :SOURce ] :PULM :EXTernal :POLarity NORMAL | INVerted  
[ :SOURce ] :PULM :EXTernal :POLarity ?
```

This command selects the polarity of the TTL input signal at the GATE/PULSE/TRIGGER INPUT front panel connector. The signal generator can respond to either a normal (a TTL high) or an inverted (TTL low) signal.

#### Example

```
:PULM:EXT:POL NORM
```

The preceding example selects normal (TTL high) polarity.

**\*RST** Normal

**Key Entry** Ext Polarity Normal Inverted

### :PULM:INTernal[1]:DElay

**Supported** All models with Option UNU or UNW

```
[ :SOURce ] :PULM :INTernal [ 1 ] :DElay <num><time_suffix> | UP | DOWN  
[ :SOURce ] :PULM :INTernal [ 1 ] :DElay ?
```

This command sets the pulse delay for the internally generated pulse modulation using the variable <num>[<time\_suffix>]. The command, used with the UP|DOWN parameters, will change the delay by a user-defined step value. Refer to the [:PULM:INTernal\[1\]:DElay:STEP](#) command for setting the value associated with the UP and DOWN choices.

The optional variable <time\_suffix> accepts nS (nanoseconds) to S (seconds).

The range value is dependent on the pulse period. Refer to the [:PULM:INTernal\[1\]:PERiod](#) command for pulse period settings.

#### Example

```
:PULM:INT:DEL 200E-9
```

The preceding example sets the internal pulse delay to 200 nanoseconds.

**\*RST** +0.00000000E+000

**Range** *Internal Free Run:* depends on pulse period and pulse width settings  
*Internal Triggered & Doublet:* 70 nS to (42 S - 20 nS - pulse width)

**Key Entry** Pulse Delay

## **:PULM:INTernal[1]:DELay:STEP**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:INTernal[1]:DELay:STEP <num><time_suffix>
[:SOURCE]:PULM:INTernal[1]:DELay:STEP?
```

This command sets the step increment for the pulse delay.

The step value set by this command is used with the UP and DOWN choices in the [:PULM:INTernal\[1\]:DELay](#) command.

The step value set with this command is not affected by a signal generator power-on, preset, or \*RST command.

### **Example**

```
:PULM:INT:DEL:STEP 10NS
```

The preceding example sets the pulse delay step value to 10 nanoseconds.

**Range** 10 nS to (pulse period - 20 nS)

## **:PULM:INTernal[1]:FREQuency**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:INTernal[1]:FREQuency <val><units>|UP|DOWN
[:SOURCE]:PULM:INTernal[1]:FREQuency?
```

This command sets the pulse rate for the internally generated square wave using the variable <val><units>. The command, used with the UP|DOWN parameters, will change the frequency by a user-defined step value. Refer to the [:PULM:INTernal\[1\]:FREQuency:STEP](#) command for setting the value associated with the UP and DOWN choices.

This command is used when SQUARE is the pulse modulation type. Refer to [:PULM:SOURce](#) command for the pulse modulation type selection.

### **Example**

```
:PULM:INT:FREQ 1MHZ
```

The preceding example sets the square wave pulse rate to 1 megahertz.

**\*RST** +4.00000000E+002

**Range** 0.1 Hz to 10 MHz

**Key Entry** **Pulse Rate**

## **:PULM:INTernal[1]:FREQuency:STEP**

**Supported** All models with Option UNU or UNW

```
[ :SOURce ] :PULM:INTernal[1]:FREQuency:STEP[ :INCRement ] <frequency>  
[ :SOURce ] :PULM:INTernal[1]:FREQuency:STEP[ INCRement ]?
```

This command sets the step value for the internally generated square wave pulse rate.

This command is used when SQUARE is the pulse modulation type. For the pulse modulation type selection, refer to the [:PULM:SOURce](#) command. The step value, set with this command, is used with the UP and DOWN choices in the [:PULM:INTernal\[1\]:FREQuency](#) command.

The step value set with this command is not affected by a power-on, preset, or \*RST command.

### **Example**

```
:PULM:INT:FREQ:STEP MIN
```

The preceding example sets the step value for the square wave pulse rate to 0.1 Hz.

**Range** 0.1 Hz to 10 MHz

## **:PULM:INTernal[1]:PERiod**

**Supported** All models with Option UNU or UNW

```
[ :SOURce ] :PULM:INTernal[1]:PERiod <val><units> | UP | DOWN  
[ :SOURce ] :PULM:INTernal[1]:PERiod?
```

This command sets the pulse period for the internally generated pulse modulation using the variables <val><units>. The command, used with the UP|DOWN parameters, will change the pulse period by a user-defined step value. Refer to the [:PULM:INTernal\[1\]:PERiod:STEP\[:INCRement\]](#) command for setting the value associated with the UP and DOWN choices.

If the entered value for the pulse period is equal to or less than the value for the pulse width, the pulse width changes to a value that is less than the pulse period. Refer to the [:PULM:INTernal\[1\]:PWIDTH](#) command for setting the pulse width.

### **Example**

```
:PULM:INT:PER .5S
```

The preceding example sets the period of the internally generated pulse to 500 milliseconds.

**\*RST** +2.00000000E-006

**Range** 70 nanoseconds (nS) to 42 seconds

**Key Entry** **Pulse Period**



## **:PULM:INTernal[1]:PERiod:STEP[:INCRement]**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:INTernal[1]:PERiod:STEP[: INCRement]<val><units>|MAXimum|
MINimum|DEFault
[:SOURCE]:PULM:INTernal[1]:PERiod:STEP[: INCRement]?
```

This command sets the step value for the internal pulse period using the variable <val><units>. The step value, set with this command, is used with the UP and DOWN choices available in the [:PULM:INTernal\[1\]:PERiod](#) command.

The step value set with this command is not affected by a power-on, preset, or \*RST command.

### **Example**

```
:PULM:INT:PER:STEP .1S
```

The preceding example sets the square wave pulse rate to 100 milliseconds.

**\*RST** +1.00000000E-006

**Range** 10 nS to 42 seconds

## **:PULM:INTernal[1]:PWIDth**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:INTernal[1]:PWIDth <num><time_suffix>|UP|DOWN
[:SOURCE]:PULM:INTernal[1]:PWIDth?
```

This command sets the pulse width for the internally generated pulse signal.

This command sets the pulse width for the internally generated pulse modulation using the variables <num><time\_suffix>. The command, used with the UP|DOWN parameters, will change the pulse width by a user-defined step value. Refer to the [:PULM:INTernal\[1\]:PWIDth:STEP](#) command for setting the value associated with the UP and DOWN choices.

If the entered value for the pulse width is equal to or greater than the value for the pulse period, the pulse width changes to a value that is less than the pulse period. For more information, refer to [“:PULM:INTernal\[1\]:PERiod” on page 192](#).

---

**NOTE** A power search is recommended for signals with pulse widths less than one microsecond. Refer to [“:ALC:SEARCh” on page 143](#).

---

### **Example**

```
:PULM:INT:PWIDth 100MS
```

The preceding example sets the pulse width to 100 milliseconds.

**\*RST** +1.00000000E-006

**Range** 10 nS to (pulse period - 20 nS)

**Key Entry** **Pulse Width**

## **:PULM:INTernal[1]:PWIDth:STEP**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE ] :PULM:INTernal[1]:PWIDth:STEP <num> <time_suffix> | MAXimum | MINimum | DEFault  
[ :SOURCE ] :PULM:INTernal[1]:PWIDth:STEP?
```

This command sets the step increment for the pulse width using the variable <num><time\_suffix>.

The step value; set by this command, is used with the UP and DOWN choices available in the [:PULM:INTernal\[1\]:PWIDth](#) command.

The step value, set with this command, is not affected by a power-on, preset, or \*RST command.

### **Example**

```
:PULM:INT:PWID:STEP 100NS
```

The preceding example sets the pulse width step to 100 nanoseconds.

**\*RST** +1.00000000E-006

**Range** 10 nS to (pulse period - 20 nS)

## **:PULM:INTernal**

**Supported** All models with Option UNU or UNW

```
[ :SOURCE ] :PULM:SOURce:INTernal SQUARE | FRUN | TRIGgered | DOUBlet | GATEd  
[ :SOURCE ] :PULM:SOURce:INTernal?
```

This command selects one of the five internally generated modulation inputs. There are two external sources; Scalar and Ext Pulse, which are selected by using the [:PULM:SOURce](#) command.

### **Example**

```
:PULM:SOUR:INT SQU
```

The preceding example selects the internally generated square wave pulse modulation format.

**\*RST** FRUN (Int Free-Run)

<b>Key Entry</b>	<b>Internal Square</b>	<b>Int Free-Run</b>	<b>Int Triggered</b>	<b>Int Doublet</b>	<b>Int Gated</b>
------------------	------------------------	---------------------	----------------------	--------------------	------------------

## :PULM:SOURce

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:SOURce INTERNAL|EXTERNAL|SCALAR
[:SOURCE]:PULM:SOURce?
```

This command sets the source for pulse modulation. The INTERNAL selection accesses one of the five internally generated modulation inputs while EXTERNAL selects an external pulse (Ext Pulse) and SCALAR selects input from a scalar network analyzer.

### Example

```
:PULM:SOUR INT
```

The preceding example selects the internal free-run, pulse modulation source.

**\*RST** FRUN (Int Free-Run)

<b>Key Entry</b>	<b>Internal Square</b>	<b>Int Free-Run</b>	<b>Int Triggered</b>	<b>Int Doublet</b>	<b>Int Gated</b>
	<b>Ext Pulse</b>	<b>Scalar</b>			

## :PULM:STATe

**Supported** All models with Option UNU or UNW

```
[ :SOURCE]:PULM:STATe ON|OFF|1|0
[:SOURCE]:PULM:STATe?
```

This command enables or disables pulse modulation for the selected path.

When pulse modulation is enabled, the PULSE annunciator appears on the signal generator's front panel display.

### Example

```
:PULM:STAT ON
```

The preceding example enables the pulse modulation.

**\*RST** 0

**Key Entry** **Pulse Off On**



---

## 5 Digital Modulation Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to the E8267D PSG Vector signal generator:

- “All Subsystem–Option 601 and 602 ([:SOURCE])” on page 197
- “AWGN ARB Subsystem–Option 403 ([:SOURCE]:RADio:AWGN:ARB)” on page 198
- “AWGN Real–Time Subsystem–Option 403 ([:SOURCE]:RADio:AWGN:RT)” on page 205
- “Custom Subsystem–Option 601 and 602 ([:SOURCE]:RADio:CUSTom)” on page 206
- “Digital Modulation Subsystem ([:SOURCE]:DM)” on page 231
- “Dual ARB Subsystem–Option 601 or 602 ([:SOURCE]:RADio:ARB)” on page 247
- “Dmodulation Subsystem–Option 601 or 602 ([:SOURCE]:RADio:DMODulation:ARB)” on page 276
- “Multitone Subsystem–Option 601 or 602 ([:SOURCE]:RADio:MTONE:ARB)” on page 298
- “Real Time GPS Subsystem–Option 409 ([:SOURCE]:RADio[1]|2|3|4:GPS)” on page 311
- “Real Time MSGPS Subsystem–Option 409 ([:SOURCE]:RADio[1]|2|3|4:MSGPs)” on page 318
- “Two Tone Subsystem ([:SOURCE]:RADio:TTONE:ARB)” on page 321
- “Wideband Digital Modulation Subsystem ([:SOURCE]:WDM)” on page 330

### All Subsystem–Option 601 and 602 ([:SOURCE])

#### :RADio:ALL:OFF

**Supported** E8267D with Option 601 or 602

```
[:SOURCE]:RADio:ALL:OFF
```

This command disables all digital modulation personalities on a particular baseband. This command does not affect analog modulation.

## AWGN ARB Subsystem—Option 403 ([:SOURce]:RADio:AWGN:ARB)

### :BWIDth

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:BWIDth <val>  
[:SOURce]:RADio:AWGN:ARB:BWIDth?
```

This command adjusts the bandwidth of the AWGN waveform.

The variable <val> is expressed in units of hertz (Hz to MHz).

**\*RST** +1.00000000E+006

**Range** 5E4 to 1.5E7

**Key Entry** **Bandwidth**

### :IQ:EXtErnal:FILTer

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:IQ:EXtErnal:FILTer 40e6|THRough  
[:SOURce]:RADio:AWGN:ARB:IQ:EXtErnal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear panel I and Q outputs. Selecting a filter setting with this command will automatically set the

[:IQ:EXtErnal:FILTer:AUTO](#) command to OFF.

40e6 This choice applies a 40 MHz baseband filter.

THRough This choice bypasses filtering.

**\*RST** THR

**Key Entry** **40.000 MHz Through**

### :IQ:EXtErnal:FILTer:AUTO

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:IQ:EXtErnal:FILTer:AUTO ON|OFF|1|0  
[:SOURce]:RADio:AWGN:ARB:IQ:EXtErnal:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals routed to the rear panel I/Q outputs.

ON(1) This choice will automatically select a digital modulation filter optimized for the current signal generator settings.

OFF(0) This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to the [:IQ:EXtErnal:FILTer](#) command for selecting a filter or through path.

**\*RST** ON

**Key Entry** **I/Q Output Filter Manual Auto**

## :HEADer:CLEar

**Supported** All with Option 403

[ :SOURCE ]:RADIO:AWGN:ARB:HEADer:CLEar

This command clears the header information from the header file used by this modulation format. The **AWGN Off On** softkey must be set to On for this command to function.

**Key Entry** Clear Header

## :HEADer:SAVE

**Supported** All with Option 403

[ :SOURCE ]:RADIO:AWGN:ARB:HEADer:SAVE

This command saves the header information to the header file used by this modulation format. The **AWGN Off On** softkey must be set to On for this command to function.

**Key Entry** Save Setup To Header

## :IQ:MODulation:ATTen

**Supported** All with Option 403

[ :SOURCE ]:RADIO:AWGN:ARB:IQ:MODulation:ATTen <val>  
 [ :SOURCE ]:RADIO:AWGN:ARB:IQ:MODulation:ATTen?

This command attenuates the I/Q signals being modulated through the signal generator's RF path. The variable <val> is expressed in units of decibels (dB).

**\*RST** +2.00000000E+000

**Range** 0 to 40

**Key Entry** Modulator Atten Manual Auto

## :IQ:MODulation:ATTen:AUTO

**Supported** All with Option 403

[ :SOURCE ]:RADIO:AWGN:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0  
 [ :SOURCE ]:RADIO:AWGN:ARB:IQ:MODulation:ATTen:AUTO?

This command enables or disables the I/Q attenuation auto mode.

ON (1) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.

OFF (0) This choice holds the attenuator at its current setting or at a selected value. For setting the attenuation value, refer to the [:IQ:MODulation:ATTen](#) command.

**\*RST** 1

**Key Entry** Modulator Atten Manual Auto

## **:IQ:MODulation:FILTer**

**Supported** All with Option 403

```
[ :SOURce ] :RADio :AWGN :ARB :IQ :MODulation :FILTer 40e6 | THROugh  
[ :SOURce ] :RADio :AWGN :ARB :IQ :MODulation :FILTer ?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter with this command will automatically set the **:IQ:MODulation:ATTen:AUTO** command to OFF.

**40E6** This choice applies a 40 MHz baseband filter to the I/Q signals.

**THROugh** This choice bypasses filtering.

**\*RST** THR

**Key Entry** **40.000 MHz Through**

## **:IQ:MODulation:FILTer:AUTO**

**Supported** All with Option 403

```
[ :SOURce ] :RADio :AWGN :ARB :IQ :MODulation :FILTer :AUTO ON | OFF | 1 | 0  
[ :SOURce ] :RADio :AWGN :ARB :IQ :MODulation :FILTer :AUTO ?
```

This command enables or disables the automatic selection of the filters for I/Q signals modulated onto the RF carrier.

**ON(1)** This choice will automatically select a digital modulation filter.

**OFF(0)** This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to the **:IQ:MODulation:FILTer** command for selecting a filter or through path.

**\*RST** 1

**Key Entry** **I/Q Mod Filter Manual Auto**

## **:MDEStination:AAMPlitude**

**Supported** All with Option 403

```
[ :SOURce ] :RADio :AWGN :ARB :MDEStination :AAMPlitude NONE | M1 | M2 | M3 | M4  
[ :SOURce ] :RADio :AWGN :ARB :MDEStination :AAMPlitude ?
```

This command routes the selected marker to the Alternate Amplitude function. The NONE parameter clears the marker for the Alternate Amplitude function.

**\*RST** NONE

**Key Entry** **None Marker 1 Marker 2 Marker 3 Marker 4**



## :MDEStination:ALCHold

**Supported** All with Option 403

---

**CAUTION** Incorrect automatic level control (ALC) sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADIO:AWGN:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4
[:SOURCE]:RADIO:AWGN:ARB:MDEStination:ALCHold?
```

This command enables or disables the marker ALC hold function for the selected marker. For setting markers, see “:MARKer:[SET]” on page 256.

Use the ALC hold function when you have a waveform signal that uses idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC leveling circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker’s polarity, see “:MPOLarity:MARKer1|2|3|4” on page 260.

---

**NOTE** Do not use the ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings.

---

For more information on the marker ALC hold function, see the *User’s Guide*. For setting the marker points, see “:MARKer:[SET]” on page 256.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
*RST	NONE

### Example

```
:RAD:AWGN:ARB:MDES:ALCH M1
```

The preceding example routes marker 1 to the ALC Hold function.

**Key Entry**      **None**   **Marker 1**   **Marker 2**   **Marker 3**   **Marker 4**

## :MDEStination:PULSe

**Supported** All with Option 403

---

**CAUTION** The pulse function incorporates the ALC hold. Incorrect automatic level control (ALC) sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:AWGN:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4  
[:SOURCE]:RADio:AWGN:ARB:MDEStination:PULSe?
```

This command enables or disables the marker pulse/RF blanking function for the selected marker. The function automatically uses the ALC hold function, so there is no need to select both ALC hold and marker pulse/RF blanking functions for the same marker

---

**NOTE** Do not use ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

---

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker's polarity, see [“:MPOLarity:MARKer1|2|3|4” on page 260](#).

---

**NOTE** Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This causes either no RF output or a continuous RF output. For setting the marker points, see [“:MARKer:\[SET\]” on page 256](#).

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin. The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking.

---

For more information on the marker RF blanking function, refer to the *User's Guide*.

NONE This terminates the marker RF blanking/pulse function.

M1–M4 These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

### Example

```
:RAD:AWGN:ARB:MDES:PULS M2
```

The preceding example routes marker 2 to Pulse/RF Blanking.

**\*RST** NONE  
**Key Entry** None Marker 1 Marker 2 Marker 3 Marker 4

**:MPOLarity:MARKer1|2|3|4**

**Supported** All with Option 403

[ :SOURCE]:RADIO:AWGN:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive  
 [ :SOURCE]:RADIO:AWGN:ARB:MPOLarity:MARKer1|2|3|4?

This command sets the polarity for the selected marker. For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points.

**Example**

:RAD:AWGN:ARB:MPOL:MARK3 NEG

The preceding example sets the polarity for marker 3 to negative.

**\*RST** POS  
**Key Entry** Marker 1 Polarity Neg Pos Marker 2 Polarity Neg Pos Marker 3 Polarity Neg Pos  
 Marker 4 Polarity Neg Pos

**:LENGth**

**Supported** All with Option 403

[ :SOURCE]:RADIO:AWGN:ARB:LENGth 1048576|524288|262144|131072|65536|  
 32768|16384  
 [ :SOURCE]:RADIO:AWGN:ARB:LENGth?

This command specifies the length (number of points) of the AWGN waveform. A longer waveform yields a statistically more correct waveform.

**\*RST** 524288  
**Key Entry** 1048576 524288 262144 131072 65536 32768 16384

## :REFerence:EXTernal:FREQuency

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:REFerence:EXTernal:FREQuency <val>  
[:SOURce]:RADio:AWGN:ARB:REFerence:EXTernal:FREQuency?
```

This command allows you to enter the frequency of the applied external reference. The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear panel connector. To specify external as the ARB reference source type, refer to “:REFerence[:SOURce]” on page 285.

The variable <val> is expressed in units of hertz (Hz–MHz).

**\*RST** +1.00000000E+007

**Range** 2.5E5 to 1E8

**Key Entry** Reference Freq

## :REFerence[:SOURce]

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:REFerence[:SOURce] INTernal|EXTernal  
[:SOURce]:RADio:AWGN:ARB:REFerence[:SOURce]?
```

This command selects either an internal or external reference for the waveform clock. If the EXTernal choice is selected, the external frequency value *must* be entered and the signal must be applied to the BASEBAND GEN REF IN rear panel connector. To enter the external reference frequency, refer to “:REFerence:EXTernal:FREQuency” on page 285.

**\*RST** INT

**Key Entry** ARB Reference Ext Int

## :SCLock:RATE

**Supported** All with Option 403

```
[ :SOURce]:RADio:AWGN:ARB:SCLock:RATE <val>  
[:SOURce]:RADio:AWGN:ARB:SCLock:RATE?
```

This command sets the sample clock rate for the AWGN modulation format. The modulation format should be active before executing this command. If this command is executed before the modulation format is active, the entered value will be overridden by a calculated factory default value. To activate the modulation format, refer to “:BURSt:SHAPE:FALL:DELay” on page 208.

The variable <val> is expressed in units of hertz.

**\*RST** +1.00000000E+008

**Range** 1 to 1E8

**Key Entry** ARB Sample Clock

## :SEED

**Supported** All with Option 403

```
[:SOURce]:RADio:AWGN:ARB:SEED FIXed|RANdOm
[:SOURce]:RADio:AWGN:ARB:SEED?
```

This command toggles the AWGN waveform noise seed value type.

**FIXed** This choice selects a fixed noise seed value.

**RANdOm** This choice selects a randomly generated noise seed value.

**\*RST** FIX

**Key Entry** Noise Seed Fixed Random

## [:STATe]

**Supported** All with Option 403

```
[:SOURce]:RADio:AWGN:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:AWGN:ARB[:STATe]?
```

This command enables or disables the AWGN generator function.

**\*RST** 0

**Key Entry** Arb AWGN Off On

## AWGN Real-Time Subsystem—Option 403 ([:SOURce]:RADio:AWGN:RT)

### :BWIDth

**Supported** E8267D with Option 403

```
[:SOURce]:RADio:AWGN:RT:BWIDth <val>
[:SOURce]:RADio:AWGN:RT:BWIDth?
```

This command adjusts the real-time AWGN bandwidth value.

The variable <val> is expressed in units of hertz (Hz–MHz).

**\*RST** +1.00000000E+006

**Range** 5E4 to 8E7

**Key Entry** Bandwidth

### [:STATe]

**Supported** E8267D with Option 403

```
[:SOURce]:RADio:AWGN:RT[:STATe] ON|OFF|1|0
[:SOURce]:RADio:AWGN:RT[:STATe]?
```

This command enables or disables the operating state of real-time AWGN.

**\*RST** 0

**Key Entry** Real-time AWGN Off On

## Custom Subsystem—Option 601 and 602 ([:SOURce]:RADio:CUSTom)

### :ALPha

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:ALPha <val>  
[:SOURce]:RADio:CUSTom:ALPha?
```

This command changes the Nyquist or root Nyquist filter's alpha value. The filter alpha value can be set to a minimum level (0), a maximum level (1), or in between by using fractional numeric values (0.001–0.999). To change the current filter type, refer to “:FILTer” on page 218.

#### Example

```
:RAD:CUST:ALPH .65
```

The preceding example sets the filter alpha to .65.

**\*RST** +3.50000000E-001

**Range** 0.000 to 1.000

**Key Entry** Filter Alpha

### :BBCLock

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BBCLock INT[1]|EXT[1]  
[:SOURce]:RADio:CUSTom:BBCLock?
```

This command toggles the data (bit) clock input to the baseband generator board to either internal or external. This command is independent in each mode and works for both non-burst (continuous) and burst modes. This allows for a matrix of selections between burst/non-burst, internal/external data generation, internal/external data clock, and external bit/symbol data clock.

INT [1] This choice selects the signal generator internal data clock.

EXT [1] This choice selects an external data clock input.

A data clock or continuous symbol sync input must be supplied when external mode is used. This is ignored if the external reference is set to EXTERNAL (Refer to “:EREFerence” on page 217).

#### Example

```
:RAD:CUST:BBCL 1
```

The preceding example selects the signal generator's internal data clock.

**\*RST** INT

**Key Entry** BBG Data Clock Ext Int

## :BBT

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BBT <val>
[:SOURce]:RADio:CUSTom:BBT?
```

This command changes the bandwidth–multiplied–by–bit–time (BbT) filter parameter. The filter BbT value can be set to the maximum level (1) or in between the minimum level (0.100) and maximum level by using fractional numeric values (0.101–0.999). This command is effective only after choosing a Gaussian filter. It does not effect other types of filters (Refer to “:FILTer” on page 218).

### Example

```
:RAD:CUST:BBT .300
```

The preceding example selects a 0.300 BbT gaussian filter.

**\*RST** +5.00000000E–001

**Range** 0.100 to 1.000

**Key Entry** Filter BbT

## :BRATe

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BRATe <val>
[:SOURce]:RADio:CUSTom:BRATe?
```

This command sets the bit rate. The variable <val> is expressed in bits per second (bps–Mbps) and the maximum range value depends on the data source (internal or external), the modulation type, and filter.

When user–defined filters are selected using the :FILTer command, the upper symbol rate will be restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Msps
- FIR filter length > 16 symbols: upper limit is 25 Msps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated and will impact the relative timing of the modulated data, as well as the actual filter response. A change in the bit rate value effects the symbol rate value. Refer to the :SRATe command for more information.

To change the modulation type, refer to “:MODulation[:TYPE]” on page 221.

**Example**

:RAD:CUST:BRAT 10MBPS

The preceding example sets the bit rate to 10 megabits per second.

\*RST +4.86000000E+004

Range	<i>Modulation Type</i>	<i>Bits per Symbol</i>	<i>Internal Data</i>	<i>External Serial Data</i>
	BPSK	1	45 bps–50 Mbps	45 bps–50 Mbps
	FSK2			
	MSK			
	C4FM	2	90 bps–100 Mbps	45 bps–50 Mbps
	FSK4			
	OQPSK			
	OQPSK195			
	P4QPPSK			
	QAM4			
	QPSK			
	QPSKIS95			
	QPSKISAT			
	D8PSK	3	135 bps–150 Mbps	45 bps–50 Mbps
	EDGE			
	FSK8			
	PSK8			
	FSK16	4	180 bps–200 Mbps	45 bps–50 Mbps
	PSK16			
	QAM16			
	QAM32	5	225 bps–250 Mbps	45 bps–50 Mbps
	QAM64	6	270 bps–300 Mbps	45 bps–50 Mbps
	QAM128	7	315 bps–350 Mbps	45 bps–50 Mbps
	QAM256	8	360 bps–400 Mbps	45 bps–50 Mbps

**:BURSt:SHAPE:FALL:DELay**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ] :RADio :CUSTom :BURSt :SHAPE :FALL :DELay <val>

[ :SOURCE ] :RADio :CUSTom :BURSt :SHAPE :FALL :DELay ?

This command sets the burst shape fall delay. The variable <val> is expressed in bits with 1 bit = 1/(symbol\_rate\*bits\_per\_symbol).

To change the modulation type, refer to the :MODulation[:TYPE] command. Refer to the :SRATE command for a list of the minimum and maximum symbol rate values.



The `:BURSt:SHAPE:FDElay` command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

**Example**

```
:RAD:CUST:BURS:SHAP:FALL:DEL 50
```

The preceding example sets a 50 bit fall delay.

**\*RST** +0.00000000E+000

**Range** -22.3750 to 99

**Key Entry** **Fall Delay**

**:BURSt:SHAPE:FALL:TIME**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:BURSt:SHAPE:FALL:TIME <val>
```

```
[ :SOURCE]:RADIO:CUSTOM:BURSt:SHAPE:FALL:TIME?
```

This command sets the burst shape fall time. The variable <val> is expressed in bits with  $1 \text{ bit} = 1/(\text{symbol\_rate} \times \text{bits\_per\_symbol})$ .

To change the modulation type, refer to the `:MODulation[:TYPE]` command. Refer to the `:SRATE` command for a list of the minimum and maximum symbol rate values.

The `:BURSt:SHAPE:FTIME` command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

**Example**

```
:RAD:CUST:BURS:SHAP:FALL:TIME 100
```

The preceding example sets a 100 bit fall delay.

**\*RST** +1.00000000E+001

**Range** 0.1250 to 255.8750

**Key Entry** **Fall Time**

**:BURSt:SHAPE:FDElay**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:BURSt:SHAPE:FDElay <val>
```

```
[ :SOURCE]:RADIO:CUSTOM:BURSt:SHAPE:FDElay?
```

This command sets the burst shape fall delay. The variable <val> is expressed in bits with  $1 \text{ bit} = 1/(\text{symbol\_rate} \times \text{bits\_per\_symbol})$ .

To change the modulation type, refer to the `:MODulation[:TYPE]` command. Refer to `:SRATE` command for a list of the minimum and maximum symbol rate values.

The `:BURSt:SHAPE:FALL:DELay` command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

**Example**

```
:RAD:CUST:BURS:SHAP:FDEL 45
```

The preceding example sets a 45 bit fall delay.

**\*RST** +0.00000000E+000

**Range** -22.3750 to 99

**Key Entry** **Fall Delay**

**:BURSt:SHAPE:FTIME**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPE:FTIME <val>
```

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPE:FTIME?
```

This command sets the burst shape fall time. The variable <val> is expressed in bits with  $1 \text{ bit} = 1/(\text{symbol\_rate} * \text{bits\_per\_symbol})$ .

To change the modulation type, refer to the `:MODulation[:TYPE]` command. Refer to `:SRATE` command for a list of the minimum and maximum symbol rate values.

The `:BURSt:SHAPE:FALL:TIME` command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

**Example**

```
:RAD:CUST:BURS:SHAP:FTIM 20
```

The preceding example sets a 20 bit fall delay.

**\*RST** +0.00000000E+000

**Range** 0.1250 to 255.8750

**Key Entry** **Fall Time**

## **:BURSt:SHAPe:RDELay**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPe:RDELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RDELay?
```

This command sets the burst shape rise delay. The variable <val> is expressed in bits with 1 bit = 1/(symbol\_rate\*bits\_per\_symbol).

To change the modulation type, refer to the **:MODulation[:TYPE]** command. Refer to **:SRATE** command for a list of the minimum and maximum symbol rate values.

The **:BURSt:SHAPe:RISE:DELay** command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

### **Example**

```
:RAD:CUST:BURS:SHAP:RDEL -10
```

The preceding example sets a -10 bit rise delay.

**\*RST** +0.00000000E+000

**Range** -17.3750 to 99

**Key Entry** **Rise Delay**

## **:BURSt:SHAPe:RISE:DELay**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:DELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:DELay?
```

This command sets the burst shape rise delay. The variable <val> is expressed in bits with 1 bit = 1/(symbol\_rate\*bits\_per\_symbol).

To change the modulation type, refer to the **:MODulation[:TYPE]** command. For a list of the minimum and maximum symbol rate values, refer to the **:SRATE** command.

The **:BURSt:SHAPe:RDELay** command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

### **Example**

```
:RAD:CUST:BURS:SHAP:RISE:DEL 10
```

The preceding example sets a 10 bit rise delay.

**\*RST** +0.00000000E+000

**Range** -17.3750 to 99

**Key Entry** **Rise Delay**

## **:BURSt:SHAPE:RISE:TIME**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPE:RISE:TIME <val>  
[:SOURce]:RADio:CUSTom:BURSt:SHAPE:RISE:TIME?
```

This command sets the burst shape rise time. The variable <val> is expressed in bits with 1 bit = 1/(symbol\_rate\*bits\_per\_symbol).

To change the modulation type, refer to the **:MODulation[:TYPE]** command. For a list of the minimum and maximum symbol rate values, refer to the **:SRATE** command.

The **:BURSt:SHAPE:RTIME** command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

### **Example**

```
:RAD:CUST:BURS:SHAP:RISE:TIME .5
```

The preceding example sets a .5 bit rise delay.

<b>*RST</b>	+1.00000000E+001
<b>Range</b>	0.1250 to 121.5000
<b>Key Entry</b>	<b>Rise Time</b>

## **:BURSt:SHAPE:RTIME**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:BURSt:SHAPE:RTIME <val>  
[:SOURce]:RADio:CUSTom:BURSt:SHAPE:RTIME?
```

This command sets the burst shape rise time. The variable <val> is expressed in bits with 1 bit = 1/(symbol\_rate\*bits\_per\_symbol).

To change the modulation type, refer to the **:MODulation[:TYPE]** command. For a list of the minimum and maximum symbol rate values, refer to the **:SRATE** command.

The **:BURSt:SHAPE:RISE:TIME** command performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *User's Guide*.

### **Example**

```
:RAD:CUST:BURS:SHAP:RTIM 100
```

The preceding example sets a 100 bit rise time.

<b>*RST</b>	+1.00000000E+001
<b>Range</b>	0.1250 to 121.5000
<b>Key Entry</b>	<b>Rise Time</b>

## **:BURSt:SHAPE[:TYPE]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:BURSt:SHAPE[:TYPE] SINE | "<file_name>"
[:SOURCE]:RADIO:CUSTOM:BURSt:SHAPE[:TYPE]?
```

This command selects a user-defined or a pre-defined burst shape file.

**SINE** This choice selects the pre-defined Sine burst shape as the burst shape type.

"<file\_name>" This variable names the user burst shape file to use. Refer to [File Name Variables](#) for information on the file name syntax.

### **Example**

```
:RAD:CUST:BURSt:SHAPE "Test_File"
```

The preceding example selects a file named Test\_File from the signal generator's SHAPE directory. The directory path is implied in the command and does not need to be specified.

```
*RST SINE
```

**Key Entry**            **Sine**    **User File**

## **:CHANnel**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:CHANnel EVM|ACP
[:SOURCE]:RADIO:CUSTOM:CHANnel?
```

This command optimizes the Nyquist and root Nyquist filters to minimize error vector magnitude (EVM) or to minimize adjacent channel power (ACP).

**EVM** This choice provides the most ideal passband.

**ACP** This choice improves stopband rejection.

To change the current filter type, refer to [“:FILTer” on page 218](#).

### **Example**

```
:RAD:CUST:CHAN EVM
```

The preceding example uses EVM optimizing.

```
*RST EVM
```

**Key Entry**            **Optimize FIR for EVM ACP**

## :DACS:ALIGn

**Supported** E8267D with Option 601 or 602

```
[[:SOURce]:RADio:CUSTom:DACS:ALIGn
```

This command resets the signal generator's I/Q DAC circuitry. This operation is required any time the external VCO clock signal is lost and re-acquired or when an external VCO clock signal is first applied to the BASEBAND GEN CLK IN connector.

### Example

```
:RAD:CUST:DACS ALIG
```

The preceding example resets the I/Q DAC circuitry.

**Key Entry**            **Align DACs**

## :DATA

**Supported** E8267D with Option 601 or 602

```
[[:SOURce]:RADio:CUSTom:DATA PN9|PN11|PN15|PN20|PN23|FIX4|"<file_name>"|  
EXT|PRAM File|P4|P8|P16|P32|P64  
[:SOURce]:RADio:CUSTom:DATA?
```

This command sets the data pattern for unframed transmission. For information on the file name syntax, see [“File Name Variables” on page 10](#).

### Example

```
:RAD:CUST:DATA PN9
```

The preceding example selects a PN9 data pattern for unframed transmission.

```
*RST PN23
```

Key Entry	PN9	PN11	PN15	PN20	PN23	FIX4	User File	Ext
	4 1's & 4 0's		8 1's & 8 0's		16 1's & 16 0's		32 1's & 32 0's	
	64 1's & 64 0's		PRAM FILE					

## :DATA:EDIGitalbus

**Supported** E8267D with Option 601 or 602 and an N5102A

```
[[:SOURce]:RADio:CUSTom:DATA:EDIGitalbus:INPut:VRMS <level>  
[:SOURce]:RADio:CUSTom:DATA:EDIGitalbus:INPut:VRMS?
```

This command informs the instrument of the voltage level (Vrms) of the input signal on the digital bus. The input signal level is used to determine the I/Q attenuation.

This function is active when LVDS is on and the LVDS data direction is set to INPUT.

### Example

```
:RAD:CUST:DATA:EDIG:INP:VRMS 0.5V
```

```
:RAD:CUST:DATA:EDIG:INP:VRMS 500mV
```

The preceding examples set an rms voltage of 0.5V.

**\*RST**                    0.4V  
**Range**                    0.1 to 1.414  
**Key Entry**                DIGITAL BUS Level Vrms

### **:DATA:FIX4**

**Supported**                E8267D with Option 601 or 602

```
[:SOURCE]:RADio:CUSTom:DATA:FIX4 <val>
[:SOURCE]:RADio:CUSTom:DATA:FIX4?
```

This command sets the binary, 4-bit repeating sequence data pattern for unframed transmission according to the modulation type, symbol rate, filter, and burst shape selected for the custom modulation format. FIX4 must be selected as the data type.

<val>                    This variable is an integer value from one to 15 and represents the a four bit pattern.

#### **Example**

```
:RAD:CUST:DATA:FIX4 15
```

The preceding example selects a FIX4 data pattern consisting of four 1's.

**\*RST**                    #B0000  
**Range**                    #B0000 to #B1111 or 0 to 15  
**Key Entry**                **FIX4**

### **:DATA:PRAM**

**Supported**                E8267D with Option 601 or 602

```
[:SOURCE]:RADio:CUSTom:DATA:PRAM "<file_name>"
[:SOURCE]:RADio:CUSTom:DATA:PRAM?
```

This command selects PRAM data as the data pattern for unframed transmission. Refer to the [:DATA:PRAM:FILE:BLOCK](#) command for information on PRAM data. For information on the file name syntax, refer to “[File Name Variables](#)” on page 10.

#### **Example**

```
:RAD:CUST:DATA:PRAM "Test_Data"
```

The preceding example selects the PRAM file, Test\_Data, as the data pattern for unframed transmission.

**Key Entry**                **PRAM File**

### **:DENCodE**

**Supported**                E8267D with Option 601 or 602

```
[:SOURCE]:RADio:CUSTom:DENCodE ON|OFF|1|0
```

[ :SOURCE ]:RADio:CUSTom:DENCode?

This command enables or disables the differential data encoding function. Executing this command encodes the data bits prior to modulation; each modulated bit is 1 if the data bit is different from the previous one or 0 if the data bit is the same as the previous one.

**Example**

```
:RAD:CUST:DENC 1
```

The preceding example enables differential data encoding for the selected modulation.

```
*RST 0
```

**Key Entry            Diff Data Encode Off On**



## :EDATa:DELaY

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:EDATa:DELaY?
```

This query returns the time delay (in symbols) from the external data input to the beginning of the symbol on the I OUT and Q OUT rear panel connectors and the front panel RF OUTPUT connector. When the format is turned off, the delay value is unchanged; the query will return the same delay value if the format is on or off.

## :EDCLock

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:EDCLock SYMBol|NORMal  

[:SOURCE]:RADIO:CUSTOM:EDCLock?
```

This command sets the external data clock use. In internal clock mode, neither choice has an effect. To select EXT as the data clock type, use the [:BBCLock](#) command.

SYMBol	This choice specifies that a continuous symbol clock signal must be provided to the SYMBOL SYNC input connector.
NORMal	This choice specifies that the DATA CLOCK input connector requires a bit clock. The SYMBOL SYNC input connector requires a (one-shot or continuous) symbol sync signal.

### Example

```
:RAD:CUST:EDCL NORM
```

The preceding example selects normal mode for the external data clock type.

```
*RST NORM
```

**Key Entry**            **Ext Data Clock Normal Symbol**

## :EREFerence

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:EREFerence INTernal|EXTernal  

[:SOURCE]:RADIO:CUSTOM:EREFerence?
```

This command selects either an internal or external bit-clock reference for the data generator.

If the EXTernal choice is selected, the external frequency value must be applied to the BASEBAND GEN REF IN rear panel connector. To enter the external reference frequency, refer to the [:EREFerence:VALue](#) command.

### Example

```
:RAD:CUST:EREF EXT
```

The preceding example selects an external bit-clock reference for the data generator.

```
*RST INT
```

**Key Entry**            **BBG Ref Ext Int**

## **:EREFERENCE:VALue**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:EREFERENCE:VALue <val>  
[ :SOURCE]:RADIO:CUSTOM:EREFERENCE:VALue?
```

This command specifies the reference frequency of the externally applied reference. The variable <val> is expressed in hertz (Hz–MHz).

The value specified by this command is valid only when an external reference is applied to the BASEBAND GEN REF IN rear panel connector. To select EXTERNAL as the bit-clock reference for the data generator, use the [:EREFERENCE](#) command.

### **Example**

```
:RAD:CUST:EREF:VAL 10E6
```

The preceding example uses a 10 MHz external reference for the signal generator's baseband generator.

**\*RST**                    +1.30000000E+007

**Range**                   2.5E5 to 1E8

**Key Entry**            **Ext BBG Ref Freq**

## **:FILTer**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:FILTer RNYQuist|NYQuist|GAUSSian|RECTangle|AC4Fm|  
UGGaussian| "<User_FIR>"  
[ :SOURCE]:RADIO:CUSTOM:FILTer?
```

This command selects the pre-modulation filter type.

RNYQuist                This choice selects a root nyquist filter (root raised cosine).

NYQuist                 This choice selects a Nyquist filter (raised cosine).

GAUSSian                This choice selects a gaussian filter.

RECTangle               This choice selects a one-symbol- wide rectangular filter.

AC4Fm                    This is a pre-defined Association of Public Safety Communications Officials (APCO) specified compatible 4-level frequency modulation (C4FM) filter.

UGGaussian              This choice selects a GSM Gaussian filter with a fixed Bbt value of 0.300.

"<User\_FIR>"            This variable is any filter file stored in the signal generator's catalog of FIR files. The directory path is implied in the command and does not need to be specified. For information on the file name syntax, see ["File Name Variables"](#) on page 10.

**\*RST**                    RNYQ

### Example

```
:RAD:CUST:FILT GAUS
```

The preceding example selects a gaussian filter as the pre-modulation filter type.

<b>Key Entry</b>	<b>Root Nyquist</b>	<b>Nyquist</b>	<b>Gaussian</b>	<b>Rectangle</b>	<b>APCO 25 C4FM</b>
	<b>UN3/4 GSM Gaussian</b>	<b>User FIR</b>			

### :IQ:SCALE

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:IQ:SCALE <val>  

[:SOURCE]:RADIO:CUSTOM:IQ:SCALE?
```

This command sets the amplitude of the I/Q outputs for better adjacent channel power (ACP); lower scaling values equate to better ACP.

The variable <val> is expressed as a percentage.

### Example

```
:RAD:CUST:IQ:SCALE 50
```

The preceding example sets I/Q scaling to 50%.

<b>*RST</b>	+70
<b>Range</b>	1 to 200
<b>Key Entry</b>	<b>I/Q Scaling</b>

### :MODulation:FSK[:DEVIation]

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:MODulation:FSK[:DEVIation] <val>  

[:SOURCE]:RADIO:CUSTOM:MODulation:FSK[:DEVIation]?
```

This command sets the maximum symmetric FSK frequency deviation value.

The variable <val> is a numeric expression in hertz which specifies the spacing of the two outermost FSK tones. Additional tones are evenly spaced between the two outermost tones. The maximum range value equals the current symbol rate value multiplied by four and is limited to 20 MHz.

To change the modulation type, refer to the [:MODulation\[:TYPE\]](#) command. For a list of the minimum and maximum symbol rate values, use the [:SRATE](#) command.

Refer to the *User's Guide* for information on setting an asymmetric FSK deviation value.

### Example

```
:RAD:CUST:MOD:FSK 50KHZ
```

The preceding example sets the maximum frequency deviation to 50 kHz.

<b>*RST</b>	+4.00000000E+002
<b>Range</b>	0 to 2E7

**Key Entry**            **Freq Dev**

### **:MODulation:MSK[:PHASe]**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:MODulation:MSK[:PHASe] <val>  
[:SOURCE]:RADIO:CUSTOM:MODulation:MSK[:PHASe]?
```

This command sets the MSK phase deviation value. The variable <val> is expressed in degrees.

#### **Example**

```
:RAD:CUST:MOD:MSK 40
```

The preceding example sets the phase deviation to 40 degrees.

```
*RST                    +9.00000000E+001
```

**Range**                0 to 100

**Key Entry**            **Phase Dev**

### **:MODulation:UFSK**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:MODulation:UFSK "<file_name>"  
[:SOURCE]:RADIO:CUSTOM:MODulation:UFSK?
```

This command selects a user-defined FSK file from the signal generator's catalog of FSK files. The directory path is implied in the command and does not need to be specified. For information on the file name syntax, see [“File Name Variables” on page 10](#).

The user-defined FSK file is held in signal generator memory until the command that selects user FSK as the modulation type is sent. To change the current modulation type, use the [:MODulation\[:TYPE\]](#) command.

#### **Example**

```
:RAD:CUST:MOD:UFSK "Test_FSK"
```

The preceding example selects the file, Test\_FSK, from the catalog of FSK files.

**Key Entry**            **User FSK**

### **:MODulation:UIQ**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:MODulation:UIQ "<file_name>"  
[:SOURCE]:RADIO:CUSTOM:MODulation:UIQ?
```

This command selects a user-defined I/Q file from the signal generator's catalog of IQ files. The directory path is implied in the command and does not need to be specified. For information on the file name syntax, see [“File Name Variables” on page 10](#).

The user-defined I/Q file is held in signal generator memory until the command that selects user I/Q as the modulation type is sent. To change the current modulation type, refer to **:MODulation[:TYPE]** command.

**Example**

```
:RAD:CUST:MOD:UIQ "Test_IQ"
```

The preceding example selects the file, Test\_IQ, from the catalog of IQ files.

**Key Entry**            **User I/Q**

**:MODulation[:TYPE]**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:MODulation[:TYPE] BPSK|QPSK|IS95QPSK|GRAYQPSK|
OQPSK|IS95OQPSK|P4DQPSK|PSK8|PSK16|D8PSK|MSK|FSK2|FSK4|FSK8|FSK16|C4FM|
QAM4|QAM16|QAM32|QAM64|QAM128|QAM256
[:SOURCE]:RADIO:CUSTOM:MODulation[:TYPE]?
```

This command sets the modulation type for the Custom personality. For user-defined modulation; UIQ or UFSK, the file must first be specified using the **“:MODulation:UFSK”** or **“:MODulation:UIQ”** commands.

**Example**

```
:RAD:CUST:MOD BPSK
```

The preceding example selects binary phase shift keying (BPSK).

**\*RST**                    P4DQPSK

<b>Key Entry</b>	<b>BPSK</b>	<b>QPSK</b>	<b>IS-95 QPSK</b>	<b>Gray Coded QPSK</b>	<b>OQPSK</b>			
	<b>IS-95 OQPSK</b>	<b><math>\pi/4</math> DQPSK</b>	<b>8PSK</b>	<b>16PSK</b>	<b>D8PSK</b>	<b>MSK</b>	<b>2-Lvl FSK</b>	
	<b>4-Lvl FSK</b>	<b>8-Lvl FSK</b>	<b>16-Lvl FSK</b>	<b>C4FM</b>	<b>4QAM</b>	<b>16QAM</b>	<b>32QAM</b>	
	<b>64QAM</b>	<b>128QAM</b>	<b>256QAM</b>	<b>User I/Q</b>	<b>User FSK</b>			

**:POLarity[:ALL]**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:POLarity[:ALL] NORMal|INVerted
[:SOURCE]:RADIO:CUSTOM:POLarity[:ALL]?
```

This command sets the signal phase rotation direction.

- NORMal                    This choice selects normal clockwise phase rotation for the signal.
- INVerted                 This choice reverses the phase rotation of the signal by inverting the Q signal.

### Example

```
:RAD:CUST:POL INV
```

The preceding example selects reverse phase rotation for the internal Q signal.

```
*RST NORM
```

**Key Entry**            **Phase Polarity Normal Invert**

### :SRATe

**Supported**            E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:SRATe <val>  
[:SOURce]:RADio:CUSTom:SRATe?
```

This command sets the transmission symbol rate.

The variable <val> is expressed in symbols per second (sps–Mps) and the maximum range value is dependent upon the source of data (internal or external), the modulation type, and filter.

When user-defined filters are selected using the :FILTer command, the upper symbol rate will be restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Mps
- FIR filter length > 16 symbols: upper limit is 25 Mps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated as follows:

- Above 12.5 Mps, the FIR length will be truncated to 32 symbols
  - Above 25 Mps, the FIR length will be truncated to 16 symbols
- This will impact the relative timing of the modulated data, as well as the actual filter response.

A change in the symbol rate value effects the bit rate value.

To change the modulation type, refer to “:MODulation[:TYPE]” on page 221.

### Example

```
:RAD:CUST:SRAT 10KSPS
```

The preceding example sets the symbol rate to 10K symbols per second.

```
*RST +2.43000000E+004
```

**Range**

<i>Modulation Type</i>	<i>Bits per Symbol</i>	<i>Internal Data</i>	<i>External Serial Data</i>
BPSK	1	1 sps–50 Msp	1 sps–50 Msp
FSK2			
MSK			
C4FM	2	1 sps–50 Msp	1 sps–25 Msp
FSK4			
OQPSK			
OQPSK195			
P4QPPSK			
QAM4			
QPSK			
QPSKIS95			
QPSKISAT			
D8PSK	3	1 sps–50 Msp	1 sps–16.67 Msp
EDGE			
FSK8			
PSK8			
FSK16	4	1 sps–50 Msp	1 sps–12.5 Msp
PSK16			
QAM16			
QAM32	5	1 sps–50 Msp	1 sps–10 Msp
QAM64	6	1 sps–50 Msp	1 sps–8.33 Msp
QAM128	7	1 sps–50 Msp	1 sps–7.142857142 Msp
QAM256	8	1 sps–50 Msp	1 sps–6.25 Msp

**Key Entry**

**Symbol Rate**

**:STANDARD:SELECT**

**Supported** E8267D with Option 601 or 602

[ :SOURCE]:RADIO:CUSTOM:STANDARD:SELECT NONE|AC4Fm|ACQPsk|BLUETOOTH|CDPD  
 [:SOURCE]:RADIO:CUSTOM:STANDARD:SELECT?

This command selects a pre-defined setup for Custom (with the appropriate defaults) and/or clears the selection.

- NONE This choice clears the current pre-defined Custom format.
- AC4Fm This choice sets up an Association of Public Safety Communications Officials (APCO) compliant, compatible 4-level frequency modulation (C4FM) format.
- ACQPsk This choice sets up an Association of Public Safety Communications Officials (APCO) compliant, compatible quadrature phase shift keying (CQPSK) format.
- BLUETOOTH This choice sets up a Bluetooth (2-level frequency shift keying) format.

CDPD                    This choice sets up a minimum shift keying Cellular Digital Packet Data (CDPD) format.

### Example

```
:RAD:CUST:STAN:SEL AC4FM
```

The preceding example selects the AC4FM pre-defined operating mode.

```
*RST                    NONE
```

```
Key Entry              None    APCO 25w/C4FM    APCO 25 w/CQPSK    Bluetooth    CDPD
```

### :TRIGger:TYPE

**Supported**            E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:TRIGger:TYPE CONTInuous|SINGle|GATE  
[:SOURce]:RADio:CUSTom:TRIGger:TYPE?
```

This commands sets the trigger mode (type) that controls the data transmission.

Triggers control the data transmission by telling the PSG when to transmit the modulating signal. Depending on the trigger settings for the PSG, the data transmission can occur once, continuously, or the PSG may start and stop the transmission repeatedly (GATE mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state of the trigger signal. It is common to have multiple triggers, also referred to as trigger occurrences or events, occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the modulating signal. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the signal's final I and Q levels determine whether you see the carrier signal or not (zero = no carrier, other values = visible carrier). At the end of most data patterns, the final I and Q points are set to a value other than zero. If you create your own data file, you can set the initial I and Q voltages to values other than zero, and set the last I and Q values to zero. Create your own file using the front panel UI (refer to the *User's Guide*), or download a file you create external to the PSG (refer to the *Agilent Signal Generators Programming Guide*).

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the data transmission.
- Setting the data pattern's response to triggers:
  - CONTInuous, see “:TRIGger:TYPE:CONTInuous[:TYPE]” on page 225
  - SINGle, selecting the mode also sets the response (This differs from using the single mode for the ARB formats.)
  - GATE, selecting the mode also sets the response
- Selecting the trigger source using the :TRIGger[:SOURce] command, determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:
  - CONTInuous and SINGle, see “:TRIGger[:SOURce]:EXTernal:SLOPe” on page 229



- GATE, see “:TRIGger:TYPE:GATE:ACTive” on page 226

For more information on triggering, see the *User’s Guide*.

The following list describes the trigger type command choices:

CONTInuous	Upon triggering, the data pattern repeats continuously.
SINGLE	Upon triggering, the data pattern plays once.
GATE	An external trigger signal controls the data transmission. The modulating signal waits for the first active trigger signal state to begin. After the initial trigger, the behavior is dependent on whether the signal incorporates framed or unframed data. Because the PSG provides only unframed data for real-time custom, to transmit a framed data signal you must create an external file that incorporates the framing and download it to the PSG. The following list describes the behavior differences between the two types of data transmissions: <ul style="list-style-type: none"> <li>• For unframed data, an external trigger signal repeatedly starts and stops the data transmission. The length of each transmission depends on the duty period of the trigger signal and the gate polarity selection (see “:TRIGger:TYPE:GATE:ACTive” on page 226). Data transmits during the active polarity selection state and stops during the inactive state. The active state can be set high or low.</li> </ul>

---

**NOTE** The real-time custom gating behavior described above is opposite to the ARB gating behavior.

---

- For framed data, an external trigger signals the PSG to start transmitting at the beginning of a frame during active states, but only stops at the end of a frame when the end occurs during the inactive states. If the end of the frame extends into the next active trigger state, the signal transmits continuously. For information on downloading files, refer to the *Agilent Signal Generators Programming Guide*.

### Example

```
:RAD:CUST:TRIG:TYPE SING
```

The preceding example selects the single trigger mode for data transmission.

```
*RST          CONT
```

<b>Key Entry</b>	<b>Continuous</b>	<b>Single</b>	<b>Gated</b>
------------------	-------------------	---------------	--------------

### :TRIGger:TYPE:CONTInuous[:TYPE]

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTInuous[:TYPE] FREE|TRIGger  

[:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTInuous[:TYPE]?
```

This command selects the data pattern’s response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see “:TRIGger:TYPE” on page 224.

The following list describes the data pattern's response to each of the command choices:

- FREE**                   Turning custom on immediately triggers the modulating signal. The signal repeats the data pattern until you turn the signal off, select another trigger, or choose another data pattern.
- TRIGger**               The modulating signal waits for a trigger before transmission begins. When the signal receives the trigger, it transmits the data continuously until you turn the signal off, select another trigger, or choose another data pattern.

**Example**

```
:RAD:CUST:TRIG:TYPE:CONT FREE
```

The preceding example selects the free-run mode for continuous data transmission.

<b>*RST</b>	<b>FREE</b>	
<b>Key Entry</b>	<b>Free Run</b>	<b>Trigger &amp; Run</b>

**:TRIGger:TYPE:GATE:ACTive**

**Supported**           E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive LOW|HIGH  
[:SOURCE]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode. The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. The PSG uses the active state to transmit the data pattern. When the inactive state occurs, the transmission stops at the last transmitted symbol, then restarts at the next symbol when the active state occurs. For more information on triggering and to select gating as the trigger mode, see “:TRIGger:TYPE” on page 224.

The following list describes the PSG's gating behavior for the polarity selections:

- LOW**                   The PSG transmits the data pattern while the trigger signal is low (active state) and stops when the trigger signal goes high (inactive state).
- HIGH**                  The PSG transmits the data pattern while the trigger signal is high (active state) and stops when the trigger signal goes low (inactive state).

**Example**

```
:RAD:CUST:TRIG:TYPE:GATE:ACT HIGH
```

The preceding example selects a high external signal level as the active state for the gate trigger.

<b>*RST</b>	<b>HIGH</b>	
<b>Key Entry</b>	<b>Gate Active Low High</b>	

## **:TRIGger[:SOURce]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:TRIGger[ :SOURce] KEY|EXT|BUS
[:SOURce]:RADio:CUSTom:TRIGger[ :SOURce]?
```

This command sets the trigger source.

For more information on triggering, see “:TRIGger:TYPE” on page 224. The following list describes the command choices:

**KEY** This choice enables manual triggering by pressing the front panel **Trigger** hardkey.

**EXT** An externally applied signal triggers the modulating signal. This is the only choice that works with gating. The following settings affect an external trigger:

- The input connector for the trigger signal. You have a choice between the rear panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector. To make the connector selection, see “:TRIGger[:SOURce]:EXTernal[:SOURce]” on page 229.

For more information on the connectors and on connecting the cables, refer to the *User’s Guide*.

- The trigger signal polarity:
  - gating mode, see “:TRIGger:TYPE:GATE:ACTivE” on page 226
  - continuous and single modes, see “:TRIGger[:SOURce]:EXTernal:SLOPe” on page 229
- Any desired delay between when the PSG receives a trigger and when the data pattern responds to the trigger. There are two parts to setting the delay:
  - setting the amount of delay, see “:TRIGger[:SOURce]:EXTernal:DELay” on page 228
  - turning the delay on, see “:TRIGger[:SOURce]:EXTernal:DELay:STATe” on page 228

**BUS** This choice enables triggering over the GPIB using the \*TRG or GET command, or the LAN and the AUXILIARY INTERFACE (RS-232) using the \*TRG command.

### **Example**

```
:RAD:CUST:TRIG BUS
```

The preceding example selects BUS triggering.

```
*RST EXT
```

<b>Key Entry</b>	<b>Trigger Key</b>	<b>Ext</b>	<b>Bus</b>
------------------	--------------------	------------	------------

## **:TRIGger[:SOURce]:EXTernal:DELay**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay <val>  
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay?
```

This command sets the number of bits to delay the PSG's response to an external trigger.

The bit delay is a delay between when the PSG receives the trigger and when it responds to the trigger. The delay uses the clocks of the bit-clock to time the delay. After the PSG receives the trigger and the set number of delay bits (clocks) occurs, the PSG transmits the data pattern.

The delay occurs after you enable the state. Refer to the **:TRIGger[:SOURce]:EXTernal:DELay:STATE** command. You can set the number of bits either before or after enabling the state.

For more information on configuring an external trigger source and to select external as the trigger source, refer to the **:TRIGger[:SOURce]** command.

### **Example**

```
:RAD:CUST:TRIG:EXT:DELay 200000
```

The preceding example sets the delay for an external trigger for 200K bits.

**\*RST** +0

**Range** 0 to 1048575

**Key Entry** Ext Delay Bits

## **:TRIGger[:SOURce]:EXTernal:DELay:STATE**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay:STATE ON|OFF|1|0  
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay:STATE?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, refer to the **:TRIGger[:SOURce]:EXTernal:DELay** command; and for more information on configuring an external source, refer to the **:TRIGger[:SOURce]** command.

### **Example**

```
:RAD:CUST:TRIG:EXT:DEL:STAT 0
```

The preceding example disables the delay state for an external trigger source.

**\*RST** 0

**Key Entry** Ext Delay Off On

## **:TRIGger[:SOURCE]:EXTernal:SLOPe**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:TRIGger[:SOURCE]:EXTernal:SLOPe POSitive|NEGative  

[:SOURCE]:RADIO:CUSTOM:TRIGger[:SOURCE]:EXTernal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous or single triggering modes. To set the polarity for gating, refer to the **:TRIGger:TYPE:GATE:ACTIVE** command.

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (transmits) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, refer to the **:TRIGger[:SOURCE]** command.

### **Example**

```
:RAD:CUST:TRIG:EXT:SLOP NEG
```

The preceding example selects the negative trigger as the active state for data transmission.

```
*RST          NEG
```

**Key Entry**            **Ext Polarity Neg Pos**

## **:TRIGger[:SOURCE]:EXTernal[:SOURCE]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:CUSTOM:TRIGger[:SOURCE]:EXTernal[:SOURCE] EPT1|EPT2|  

EPTRIGGER1|EPTRIGGER2  

[:SOURCE]:RADIO:CUSTOM:TRIGger[:SOURCE]:EXTernal[:SOURCE]?
```

This command selects which rear panel connector the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, refer to the **:TRIGger[:SOURCE]** command. For more information on the rear panel connectors, refer to the *User's Guide*.

The following list describes the command choices:

EPT1	This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear panel connector.
EPT2	This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.
EPTRIGGER1	This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear panel connector.
EPTRIGGER2	This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.

**Example**

```
:RAD:CUST:TRIG:EXT EPT2
```

The preceding example selects an external trigger from the PATTERN TRIG IN 2 rear panel connector.

<b>*RST</b>	EPT1	
<b>Key Entry</b>	<b>Patt Trig In 1</b>	<b>Patt Trig In 2</b>

**[ :STATe ]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ]:RADio:CUSTom[:STATe] ON|OFF|1|0  
[:SOURce]:RADio:CUSTom[:STATe]?
```

This command enables or disables the Custom modulation format.

Although the Custom modulation is enabled with this command, the RF carrier is not modulated unless you activate the front panel **Mod On/Off** hardkey.

**Example**

```
:RAD:CUST OFF
```

The preceding example turns off the custom modulation format.

<b>*RST</b>	0
<b>Key Entry</b>	<b>Custom Off On</b>

**:VCO:CLOCK**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ]:RADio:CUSTom:VCO:CLOCK INternal|EXTernal  
[:SOURce]:RADio:CUSTom:VCO:CLOCK?
```

This command enables an internal or external VCO clock. The external VCO clock is connected to the rear panel BASEBAND GEN CLK IN connector. Use the [:DACS:ALIGN](#) command after an external VCO clock is first applied to the BASEBAND GEN CLK IN connector or when the VCO signal is lost and then re-acquired.

**Example**

```
:RAD:CUST:VCO:CLOC EXT
```

The preceding example selects an external VCO clock.

<b>*RST</b>	Int
<b>Key Entry</b>	<b>VCO Clock Ext Int</b>

## Digital Modulation Subsystem ([:SOURce]:DM)

### :EXternal:Filter

**Supported** E8267D

```
[ :SOURce ] :DM :EXternal :FILTer 40e6 | THROugh  
[ :SOURce ] :DM :EXternal :FILTer ?
```

This command selects the filter or through path for I/Q signals routed to the rear panel I and Q outputs.

40e6 This choice applies a 40 MHz baseband filter.

THROugh This choice bypasses filtering.

### Example

```
:DM :EXT :FILT 40E6
```

The preceding example selects the 40 MHz baseband filter.

**\*RST** THR

**Key Entry** 40.000 MHz Through

### :EXternal:Filter:AUTO

**Supported** E8267D

```
[ :SOURce ] :DM :EXternal :FILTer :AUTO ON | OFF | 1 | 0  
[ :SOURce ] :DM :EXternal :FILTer :AUTO ?
```

This command enables or disables the automatic filter selection for I/Q signals routed to the rear panel I/Q outputs.

ON(1) This choice automatically selects the 40 MHz filter optimized for the current signal generator settings.

OFF(0) This choice disables the auto feature and allows you to select the 40 MHz filter or a through path. Refer to the [:IQ:EXternal:Filter](#) command for selecting a filter or through path.

### Example

```
:DM :EXT :FILT :AUTO 1
```

The preceding example allows automatic selection of the 40 MHz I/Q filter.

**\*RST** 1 (ON)

**Key Entry** I/Q Output Filter Manual Auto

## **:EXternal:HCRest**

**Supported** E8267D

```
[ :SOURce]:DM:EXternal:HCRest [STATe] ON|OFF|1|0  
[:SOURce]:DM:EXternal:HCRest [STATe]?
```

This command changes the operating condition to accommodate I/Q inputs with a high crest factor.

ON (1) This choice turns high crest mode on for externally applied signals with high crest factors. High crest mode allows the signal generator to process these signals with less distortion. For crest factors higher than 4 dB, I/Q drive levels should be reduced by 1 dB for each dB above that level. In high crest mode, the maximum output level is reduced and power level accuracy is degraded.

OFF (0) This choice disables the high crest mode.

### **Example**

```
:DM:EXT:HCR 0
```

The preceding example disables the high crest mode.

**\*RST** NORM

**Key Entry** High Crest Mode Off On

## **:EXternal:POLarity**

**Supported** E8267D

```
[ :SOURce]:DM:EXternal:POLarity NORMal|INVert|INverted  
[:SOURce]:DM:EXternal:POLarity?
```

This command, for backward compatibility with older ESG E44xxB models, selects normal or inverted I/Q signal routing. In inverted mode, the Q input is routed to the I modulator and the I input is routed to the Q modulator.

### **Example**

```
:DM:EXT:POL INV
```

The preceding example inverts I and Q signal routing.

**\*RST** NORM

**Key Entry** Int Phase Polarity Normal Invert



## :EXTERNAL:SOURce

**Supported** E8267D

```
[ :SOURce]:DM:EXTernal:SOURce EXTernal|INTernal|BBG1|EXT600|OFF|SUM
[:SOURce]:DM:EXTernal:SOURce?
```

This command selects the I/Q signal source that is routed to the rear panel I and Q output connectors.

EXTernal	This choice routes a portion of the externally applied signals at the 50 ohm I and Q input connectors to the rear panel I and Q output connectors.
INTernal	This choice is for backward compatibility and performs the same function as the BBG1 selection.
BBG1	This choice routes a portion of the baseband generator I/Q signals to the rear panel I and Q connectors and requires Option 602.
EXT600	This choice routes a portion of the externally applied signals at the 600 ohm I and Q input connectors to the rear panel I and Q output connectors.
OFF	This choice disables the output to the rear panel I and Q output connectors.

The output is the analog component of the I and Q signals.

For selecting the I/Q source, refer to the :SOURce command.

### Example

```
:DM:EXT:SOUR EXT
```

The preceding example routes the I/Q signals to the external 50 ohm rear-panel output.

```
*RST EXT
Key Entry Ext 50 Ohm BBG1 Ext 600 Ohm Off
```

## :IQADjustment:DELay

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:DELay <delay_val>
[:SOURce]:DM:IQADjustment:DELay?
```

This command sets a delay for both I and Q from the baseband to the I/Q outputs and to the RF output. This will allow you to time shift the I/Q with respect to triggering and markers. The absolute phase of both I and Q will change with respect to triggers and markers. A positive value advances the I and Q phase. The range limits are dependent on the current modulation format.

This feature is not compatible with constant envelope modulations and signals connected to the external I/Q inputs.

The <delay\_val> variable is expressed in seconds.

### Example

```
:DM:IQAD:DEL .05SEC
```

The preceding example sets a 50 millisecond delay for the I and Q signals.

```
*RST +0.00000000E+000
```

**Key Entry**            **I/Q Delay**

### **:IQADjustment:EXTernal:COFFset**

**Supported**            E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:COFFset <units>  
[ :SOURce]:DM:IQADjustment:EXTernal:COFFset?
```

This command sets the common mode offset voltage for both the in-phase (I) and quadrature-phase (Q) signals going to the rear panel I and Q output connectors.

The <units> variable is expressed in volts (mV-V). This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “[:IQADjustment\[:STATe\]](#)” on page 241.

### Example

```
:DM:IQAD:EXT:COFF -.1
```

The preceding example sets a negative .1 volt common mode offset voltage for the I and Q signals.

```
*RST +0.00000000E+000
```

**Range**                -3 to 3

**Key Entry**            **Common Mode I/Q Offset**

### **:IQADjustment:EXTernal:DIOFFset**

**Supported**            E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:DIOFFset <val><units>  
[ :SOURce]:DM:IQADjustment:EXTernal:DIOFFset?
```

This command sets the differential offset voltage for an in-phase (I) signal routed to the I output connectors.

The variable <val> is a numeric expression. The <units> variable is expressed in volts (mV-V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “[:IQADjustment\[:STATe\]](#)” on page 241.

### Example

```
:DM:IQAD:EXT:DIOF 1
```

The preceding example sets a 1 volt differential offset voltage for the I signal at the rear panel I output connector.

```
*RST +0.00000000E+000
```

**Range**                -3 to 3

**Key Entry**            **Diff. Mode I Offset**

**:IQADjustment:EXTernal:DQOffset**

**Supported**            E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:DQOffset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:DQOffset?
```

This command sets the differential offset voltage for a quadrature-phase (Q) signal routed to the Q output connectors.

The variable <val> is a numeric expression. The <units> variable is expressed in volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

**Example**

```
:DM:IQAD:EXT:DQOF 1
```

The preceding example sets a 1 volt differential offset voltage for the Q signal at the rear panel Q connector.

**\*RST**                    +0.00000000E+000

**Range**                    –3 to 3

**Key Entry**            **Diff. Mode Q Offset**

**:IQADjustment:EXTernal:GAIN**

**Supported**            E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:GAIN[1|2] <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:GAIN?
```

This command sets the I/Q gain ratio (I/Q balance) for signals routed to the rear panel I and Q output connectors. The I signal (GAIN 1) is increased for positive values and the Q signal (GAIN 2) level increases with negative values

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

**Example**

```
:DM:IQAD:EXT:GAIN2 1
```

The preceding example sets a Q gain ratio of 1 volt.

**\*RST**                    +0.00000000E+000

**Range**                    –4 to 4

**Key Entry**            **I/Q Out Gain Balance**

## **:IQADjustment:EXTernal:IOFFset**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:IOFFset <val><units>  
[:SOURce]:DM:IQADjustment:EXTernal:IOFFset?
```

This command sets the offset voltage for a signal applied to the 600 ohm I input connector.

The variable <val> is a numeric expression. The <units> variable is expressed in volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

### **Example**

```
:DM:IQAD:EXT:IOFF 200MV
```

The preceding example sets a 200 millivolt offset for the signal applied to the I 600 ohm input connector.

**\*RST** +0.00000000E+000

**Range** –5 to 5

**Key Entry** Ext In 600 Ohm I Offset

## **:IQADjustment:EXTernal:IQATten**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:EXTernal:IQATten <val><units>  
[:SOURce]:DM:IQADjustment:EXTernal:IQATten?
```

This command sets the I/Q output attenuation level.

The variable <val> is a numeric expression. The <units> variable is expressed in decibels (dB).

The value set by this command is active even if the I/Q adjustment function is off.

### **Example**

```
:DM:IQAD:EXT:IQAT 10.1
```

The preceding example sets the IQ attenuator level to 10.1 dB.

**\*RST** +6.00000000E+000

**Range** 0 to 40

**Key Entry** I/Q Output Atten

## **:IQADjustment:EXTernal:QOFFset**

**Supported** E8267D

```
[ :SOURce ] :DM :IQADjustment :EXTernal :QOFFset <val><units>
[ :SOURce ] :DM :IQADjustment :EXTernal :QOFFset ?
```

This command sets the offset voltage for a signal applied to the 600 ohm Q input connector. The variable <val> is a numeric expression. The <units> variable is expressed in volts (mV-V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

### **Example**

```
:DM:IQAD:EXT:QOFF 200MV
```

The preceding example sets a 200 millivolt offset for the signal applied to the Q 600 ohm input connector.

**\*RST** +0.00000000E+000

**Range** -5 to 5

**Key Entry** Ext In 600 Ohm Q Offset

## **:IQADjustment:GAIN**

**Supported** E8267D

```
[ :SOURce ] :DM :IQADjustment :GAIN[1|2] <val>
[ :SOURce ] :DM :IQADjustment :GAIN ?
```

This command sets the gain for the I signal (GAIN 1) relative to the Q signal, (GAIN 2). The gain ratio is expressed in decibels (dB).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

### **Example**

```
:DM:IQAD:GAIN2 -3
```

The preceding example sets a gain of -3 dB for the Q signal relative to the I signal.

**\*RST** +0.00000000E+000

**Range** -4 to 4 dB

**Key Entry** I/Q Gain Balance Source 1

## **:IQADjustment:IOFFset**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:IOFFset <val>  
[:SOURce]:DM:IQADjustment:IOFFset?
```

This command adjusts the I channel offset value.

The <val> variable is expressed as a percent with 100% equivalent to 500 mV DC at the input connector. The minimum resolution is 0.025 percent.

When using this command to minimize the LO feedthrough signal, optimum performance is achieved when the command is sent after all other I/Q path commands are executed, such as those that change the internal phase polarity or adjust the modulator attenuator. If other adjustments are made after minimizing is performed, the LO feedthrough signal may increase.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

### **Example**

```
:DM:IQAD:IOFF -30
```

The preceding example sets the I channel offset to -30%.

**\*RST** +0.00000000E+000

**Range** -5E1 to +5E1

**Key Entry** **I Offset**

## **:IQADjustment:QOFFset**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:QOFFset <val>  
[:SOURce]:DM:IQADjustment:QOFFset?
```

This command adjusts the Q channel offset value.

The <val> variable is expressed as a percent with 100% equivalent to 500 mV DC at the input connector. The minimum resolution is 0.025 percent.

When using this command to minimize the LO feedthrough signal, optimum performance is achieved when the command is sent after all other I/Q path commands are executed, such as those that change the internal phase polarity or adjust the modulator attenuator. If other adjustments are made after minimizing is performed, the LO feedthrough signal may increase.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

**Example**

```
:DM:IQAD:QOFF -30
```

The preceding example sets the Q channel offset to -30%.

**\*RST** +0.00000000E+000

**Range** -5E1 to +5E1

**Key Entry** **Q Offset**

**:IQADjustment:QSKew**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:QSKew <val>
```

```
[ :SOURce]:DM:IQADjustment:QSKew?
```

This command adjusts the phase angle (quadrature skew) between the I and Q vectors by increasing or decreasing the Q phase angle.

The <val> variable is expressed in degrees with a minimum resolution of 0.1.

If the signal generator is operating at frequencies greater than 3.3 GHz, quadrature skew settings greater than ±5 degrees will not be within specifications.

Positive skew increases the angle from 90 degrees while negative skew decreases the angle from 90 degrees. When the quadrature skew is zero, the phase angle between the I and Q vectors is 90 degrees.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to “:IQADjustment[:STATe]” on page 241.

**Example**

```
:DM:IQAD:QSKew 4.5
```

The preceding example increases the phase angle by 4.5 degrees.

**\*RST** +0.00000000E+000

**Range** -1E1 to +1E1

**Key Entry** **Quadrature Angle Adjustment**

**:IQADjustment:SKEW**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment:SKEW[:DELay] <val>
```

```
[ :SOURce]:DM:IQADjustment:SKEW?
```

This command changes the input skew which is a time delay difference between the I and Q signals. Equal and opposite skew is applied to both I and Q and affects the RF Output and I/Q output paths simultaneously. A positive value delays the I signal relative to the Q signal, and a negative value delays the Q signal relative to the I signal.

If the internal I/Q correction path is set to RF or BB the I/Q signals are already optimized and adjusting I/Q skew would add an impairment to the signals. If the internal I/Q correction path is set to Off, then adjusting the I/Q skew could improve the I/Q signals. The I/Q skew adjustment cannot be performed on the MSK, FSK, and C4FM constant envelope modulations.

I/Q skew adjustments are preserved when the instrument state is saved. I/Q skew adjustments are also preserved when instrument settings are changed. If the signal generator is calibrated, the skew adjustments are added to the calibration value used for the given signal generator state. If the signal generator is uncalibrated, the skew adjustments are re-applied directly.

Using I/Q skew while playing a user FIR file greater than 32 symbols will generate an error.

The variable <val> is expressed in seconds. Range limits are determined by the modulation configuration but is limited to a maximum of  $\pm 2$  seconds.

### Example

```
:DM:IQAD:SKEW .5
```

The preceding example sets the time delay difference between the I and Q signals to 500 milliseconds.

```
*RST +0.00000000E+000
```

**Key Entry**            **I/Q Timing Skew**

### :IQADjustment:SKEW:Path

**Supported**            E8267D

```
[ :SOURce]:DM:IQADjustment:SKEW:PATH RF BB  
[:SOURce]:DM:IQADjustment:SKEW?
```

This command selects either the RF or BB (baseband) path as the path to which skew timing corrections will be applied. If there are no factory I/Q timing skew corrections data, then adjusting the I/Q timing skew for the selected path may improve the error vector magnitude (EVM) of the signal. Refer to the [:IQADjustment:SKEW](#) command for more information.

If internal I/Q corrections are available for the RF or external I/Q output (BB) path then the I/Q signals are already optimized and adjusting I/Q skew for either path would add an impairment to the signal.

### Example

```
:DM:IQAD:SKEW:PATH RF
```

The preceding example selects the RF path as the path to which skew timing adjustments will be made.

```
*RST +0.00000000E+000
```

**Key Entry**            **I/Q Timing Skew Path**



## **:IQADjustment[:STATe]**

**Supported** E8267D

```
[ :SOURce]:DM:IQADjustment[:STATe] ON|OFF|1|0  
[:SOURce]:DM:IQADjustment[:STATe]?
```

This command enables or disables the I/Q adjustments.

### **Example**

```
:DM:IQAD 1
```

The preceding example enables I/Q adjustments.

**\*RST** 0 (OFF)

**Key Entry** I/Q Adjustments Off On

## **:MODulation:ATTen**

**Supported** E8267D

```
[ :SOURce]:DM:MODulation:ATTen <val>  
[:SOURce]:DM:MODulation:ATTen?
```

This command sets the attenuation level for the I/Q signals being modulated through the signal generator RF path. The variable <val> is expressed in decibels (dB).

### **Example**

```
:DM:MOD:ATT 10
```

The preceding example sets the modulator attenuator to 10 dB.

**\*RST** +2.00000000E+000

**Range** 0 to 40 dB

**Key Entry** Modulator Atten Manual Auto

## **:MODulation:ATTen:AUTO**

**Supported** E8267D

```
[ :SOURce]:DM:MODulation:ATTen:AUTO ON|OFF|1|0  
[:SOURce]:DM:MODulation:ATTen:AUTO?
```

This command enables or disables the modulator attenuator auto mode. The auto mode will be switched to manual if the signal generator receives a AUTO OFF or AUTO 0 command.

ON (1) This choice sets the modulator attenuator to auto mode which optimizes the attenuation setting for the current signal generator settings.

OFF (0) This choice sets the attenuator to manual mode and holds the attenuator at its current setting. Refer to the [:MODulation:ATTen](#) command for setting the attenuation value.

### Example

```
:DM:MOD:ATT:AUTO OFF
```

The preceding example sets the modulator attenuator to manual mode.

```
*RST 1
```

**Key Entry**            **Modulator Atten Manual Auto**

### :MODulation:ATTen:EXTernal

**Supported**            E8267D

```
[ :SOURce]:DM:MODulation:ATTen:EXTernal DEFault|MANual|MEASure  
[:SOURce]:DM:MODulation:ATTen:EXTernal?
```

This command selects the external measurement used to set the attenuator level. The modulation attenuation must be in Auto mode and is enabled by the [:MODulation:ATTen:AUTO](#) command.

**DEFault**            Use this choice to set the external I/Q input level to the default value of 500.0 mV.

**MANual**            Use this choice to manually set the external input level. The input level is set by using the [:MODulation:ATTen:EXTernal:LEVel](#) command.

**MEASurement**        This choice uses a real-time measurement of the external input level to set the attenuator level. The measurement will be used to set the attenuator level setting and is performed by using the [:MODulation:ATTen:EXTernal:LEVel:MEASurement](#) command.

### Example

```
:DM:MOD:ATT:EXT MAN
```

The preceding example sets manual as the method for setting the external I/Q input level.

```
*RST DEFault
```

**Key Entry**            **Ext Input Level (nnn mV) Default Man Meas**

### :MODulation:ATTen:EXTernal:LEVel

**Supported**            E8267D

```
[ :SOURce]:DM:MODulation:ATTen:EXTernal:LEVel <val><volt_units>  
[:SOURce]:DM:MODulation:ATTen:EXTernal:LEVel?
```

This command sets the I/Q signal voltage level at the external I/Q inputs. The voltage level set with this command is used as the input level setting for automatic attenuation.

### Example

```
:DM:MOD:ATT:EXT:LEV 100MV
```

The preceding example sets the voltage level for the I and Q inputs to 100 millivolts.

```
*RST +4.00000000E-001
```

**Range**            .05 to 1 Volt

**Key Entry**            **I/Q Output Atten**

### **:MODulation:ATTenn:EXTernal:LEVel:MEASurement**

**Supported** E8267D

[ :SOURce]:DM:MODulation:ATTenn:EXTernal:LEVel:MEASurement

This command measures the RMS value of the external I/Q signal. The external input level must be set to **Measure**.

**Key Entry** Do External Input Level Measurement

### **:MODulation:ATTen:OPTimize:BANDwidth**

**Supported** E8267D

[ :SOURce]:DM:MODulation:ATTen:OPTimize:BANDwidth <val><rate>

[ :SOURce]:DM:MODulation:ATTen:OPTimize:BANDwidth?

This command sets the expected bandwidth of the external I/Q signal. The bandwidth set with this command be used by the modulator attenuator for level setting.

The variable <val> is a number within the range limits and the variable <rate> is expressed as samples per second (sps, ksps, or msps).

#### **Example**

```
:DM:MOD:ATT:OPT:BAND .250MSPS
```

The preceding example measures the voltage level at the external I/Q inputs.

**\*RST** +1.00000000E+006

**Range** 1E3 to 100E6

**Key Entry** Optimize for (nnn sps) Bandwidth

### **:MODulation:FILTer**

**Supported** E8267

[ :SOURce]:DM:MODulation:FILTer 40e6|THRough

[ :SOURce]:DM:MODulation:FILTer?

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter with this command automatically sets the **:MODulation:FILTer:AUTO** command to OFF.

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough This choice uses through path filtering.

#### **Example**

```
:DM:MOD:FILT 40E6
```

The preceding example selects the 40 MHz filter for I/Q signals.

**\*RST** THR

**Key Entry** 40.000MHz Through

## **:MODulation:FILTer:AUTO**

**Supported** E8267D

```
[ :SOURce]:DM:MODulation:FILTer:AUTO ON|OFF|1|0  
[:SOURce]:DM:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals modulated onto the RF carrier.

- ON (1) This choice will automatically select the optimal filter.
- OFF (0) This choice disables the automatic filter selection and allows you to select a filter or through path. Refer to the [:IQ:MODulation:FILTer](#) command for selecting a filter or through path.

### **Example**

```
:DM:MOD:FILT:AUTO 0
```

The preceding example disables the automatic filter selection for I/Q signals.

```
*RST 1
```

**Key Entry** I/Q Mod Filter Manual Auto

## **:POLarity[:ALL]**

**Supported** E8267D

```
[ :SOURce]:DM:POLarity[:ALL] NORMal|INVert|INVerted  
[:SOURce]:DM:POLarity?
```

This command selects normal or inverted I/Q signal routing. In inverted mode, the Q input is routed to the I modulator and the I input is routed to the Q modulator, inverting the phase polarity.

- NORMal This choice selects normal routing for the I and Q signals.
- INVert (ed) This choice inverts the phase polarity by routing the I signal to the Q input of the I/Q modulator and the Q signal to the I input.

### **Example**

```
:DM:POL INV
```

The preceding example swaps the I and Q routing paths.

```
*RST NORM
```

**Key Entry** Int Phase Polarity Normal Invert

## :SKEW:PATH

**Supported** E8267D

```
[ :SOURce]:DM:SKEW:PATH RF|BB
[:SOURce]:DM:SKEW:PATH?
```

This command selects the signal path that will be optimized using I/Q skew corrections. The other path may be degraded.

RF	When RF is selected, the skew is optimized for the I/Q signal applied to the RF Output. The baseband (BB) output will be functional, but the I/Q skew applied will be optimized for the RF path. When using this choice, seven symbols of latency are added to the Arb based waveform. While in real-time mode, the maximum number of user symbols for the FIR is limited to 32.
BB	When BB is selected, the skew is optimized for the I/Q signal outputs on the rear panel. The RF Output will be functional, but the I/Q skew applied will be optimized for the BB path. When using this choice, seven symbols of latency are added to the ARB based waveform. While in real-time mode, the maximum number of user symbols for the FIR is limited to 32.

---

**NOTE** You must have a skew calibration to use this command. I/Q skew corrections and calibration must be performed at an Agilent factory or service center

---

### Example

```
:DM:SKEW:PATH BB
```

The preceding example selects the baseband path for I/Q skew and calibration.

**\*RST** RF

**Key Entry** Int I/Q Skew Corrections RF BB Off

## :SKEW[:STATe]

**Supported** E8267D

```
[ :SOURce]:DM:SKEW[:STATe] ON|OFF|1|0
[:SOURce]:DM:SKEW[:STATe]?
```

This command enables or disables the I/Q skew correction function.

### Example

```
:DM:SKEW:STAT 0
```

The preceding example disables I/Q skew corrections.

**\*RST** 1

**Key Entry** Int I/Q Skew Corrections RF BB Off

## :SOURce

**Supported** E8267D

```
[ :SOURce]:DM:SOURce[1] | 2 EXTernal | INTernal | BBG1 | EXT600 | OFF  
[:SOURce]:DM:SOURce?
```

This command selects the I/Q modulator source for one of the two possible paths.

EXTernal	This choice selects an external 50 ohm source as the I/Q input to I/Q modulator.
INTernal	This choice is for backward compatibility with ESG E44xxB models and performs the same function as the BBG1 selection.
BBG1	This choice selects the baseband generator as the source for the I/Q modulator.
EXT600	This choice selects a 600 ohm impedance for the I and Q input connectors and routes the applied signals to the I/Q modulator.
OFF	This choice disables the I/Q input.

### Example

```
:DM:SOURCE1 BBG1
```

The preceding example selects BBG1, the baseband generator, as the modulation source for path 1.

```
*RST EXT
```

<b>Key Entry</b>	<b>Ext 50 Ohm</b>	<b>BBG1</b>	<b>Ext 600 Ohm</b>	<b>Off</b>
------------------	-------------------	-------------	--------------------	------------

## :SRATio

**Supported** All

```
[ :SOURce]:DM:SRATio <val><units>  
[:SOURce]:DM:SRATio?
```

This command enables you to set the power level difference (ratio) between the source one and the source two signals when the two signals are summed together. A positive ratio value reduces the amplitude for source two while a negative ratio value reduces the amplitude for source one.

The range for the summing ratio is dependent on the modulator attenuator setting for the signal generator that is summing the signals together. The minimum range is achieved when the modulator attenuator setting is zero and the maximum range is reached when the maximum attenuator value is used. The range can be calculated using the following formula:

$$\pm \text{Range} = 50 \text{ dB} + \text{Mod Atten}$$

The variable <val> is expressed as a number. The variable <units> is expressed in decibels (dB).

For setting the modulator attenuator for real-time modulation formats; refer to the [:IQ:MODulation:ATTen](#) and [:IQ:MODulation:ATTen:AUTO](#) commands. For setting the modulator attenuator for Arb modulation formats, refer to the SCPI command subsystem for the Arb format being used and find the commands that contain the command mnemonics [IQ:MODulation:ATTen](#).

**Example**

```
:DM:SRAT 3DB
```

The preceding example sets the summing ratio for source 1 and source 2 to 3 dB.

**\*RST**                    +0.00000000E+000  
**Range**                    *Min:* ± 50 dB      *Max:* ± 90 dB  
**Key Entry**                **Summing Ratio (SRC1/SRC2) x.xx dB**

**:STATe**

**Supported**                E8267D

```
[ :SOURce]:DM:STATe ON|OFF|1|0  

[:SOURce]:DM:STATe?
```

This command enables or disables the internal I/Q modulator. The signal generator I/Q annunciator is displayed when the I/Q modulator is on.

The I/Q modulator is enabled whenever a digital format is turned on.

**Example**

```
:DM:STAT OFF
```

The preceding example turns off the I/Q modulator.

**\*RST**                    0  
**Key Entry**                **I/Q Off On**

**Dual ARB Subsystem—Option 601 or 602 ([:SOURce]:RADio:ARB)**

**:BASEband:FREQuency:OFFSet**

**Supported**                E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:BASEband:FREQuency:OFFSet <value> <unit>  

[:SOURce]:RADio:ARB:BASEband:FREQuency:OFFSet?
```

This command offsets the baseband frequency relative to the modulated carrier. This feature is useful for moving the signal such that the carrier feed-through is not in the center. When the Baseband Frequency Offset is set to a value (i.e. non-zero), a phase rotator is employed which is continually shifting the phase of the baseband signal. When the Baseband Frequency Offset is returned to 0 Hz, this phase is automatically reset to zero.

The query returns the current baseband frequency offset value.

**Example**

```
:RAD:ARB:BASE:FREQ:OFFS 20 MHz
```

The preceding example sets the baseband frequency offset value to 20 MHz.

**Default**                    0 Hz

**Range** -20 MHz to 20 MHz  
**Key Entry** **Baseband Frequency Offset**

## :CLIPping

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ] :RADio :ARB :CLIPping "<file_name>" , IJQ | IORQ , <val> [ , <val> ]
```

This command sets the clipping level of the selected waveform segment to a percentage of its highest peak. The waveform must be selected before the clipping command is executed. For more information about clipping, refer to the *User's Guide*.

The variable <val> is expressed as a percentage within a 10–100% range.

IJQ This choice clips the composite I/Q waveform.

IORQ This choice clips I and Q separately. When this choice is enabled, percentage values for both I and Q must be specified.

A value of 100 percent equates to no clipping.

For information on the file name syntax, see “File Name Variables” on page 10.

### Example

```
:RAD:ARB:CLIP "ramp_test_wfm" , IJQ , 50
```

```
:RAD:ARB:CLIP "ramp_test_wfm" , IORQ , 50 , 60
```

The preceding examples clip the ramp\_test\_wfm waveform data file. The second example clips I and Q separately to 50% and 60% respectively.

```
*RST IJQ <val>: +100
```

**Range** <val>: 10–100 (0.1% resolution)  
**Key Entry** **Clipping** **Clipping Type** |I+jQ| |I|,|Q| **Clip** |I+jQ| **To**  
**Clip** |I| **To** **Clip** |Q| **To**

## :DACS:ALIGn

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ] :RADio :ARB :DACS :ALIGn
```

This command resets the signal generator's I/Q DAC circuitry. This operation is required any time the external VCO clock signal is lost and re-acquired or when an external VCO clock signal is first applied to the BASEBAND GEN CLK IN connector.

**Key Entry** **Align DACs**



## :GENerate:SINE

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ]:RADio:ARB:GENerate:SINE [ "<file_name>" ][ ,<osr> ], [ <scale> ],  
[ I|Q|IQ ]
```

This command creates a sine wave waveform file and saves it in the signal generator's volatile waveform memory (WFM1).

"<file\_name>" This variable names the file used to save the generated sine wave data.

<osr> This variable sets the oversample ratio, which must be a value that is  $\geq 4$ . If the specified over sample ratio is  $< 60$  (the minimum number of samples or I/Q points), multiple periods are generated to create a waveform with at least 60 samples. The number of periods that will be created is  $60 \div \text{<osr>}$  (quotient will round off to a whole number). A waveform with an oversample ratio  $\geq 60$  has one period.

<scale> This variable sets the scale factor for the waveform. The scale factor must be between 0–1.

I|Q|IQ The sine wave data can be applied to the I, Q, or IQ paths.

Executing this command without the "<file\_name>" variable will generate a factory default SINE\_TEST\_WFM file. When using the variable "<file\_name>" for this command, the "@" or ":" characters are not allowed.

### Example

```
:RAD:ARB:GEN:SINE "Sine_Wave",20,.5,IQ
```

The preceding example generates an IQ sine wave and saves the data to a file named Sine\_Wave. The oversampling ratio is 20, the scaling is set for 50%, and the data is applied to both the I and Q paths.

**Range** 4 to 32 Msamples (limited to available baseband memory)

## :HEADer:CLEar

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ]:RADio:ARB:HEADer:CLEar
```

This command clears the header information from the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

The dual ARB must be On for this command to function and is enabled by using the [:STATe] command.

**Key Entry** Clear Header

## :HEADer:RMS

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:HEADer:RMS <"file_name">,<val>|UNSPecified  
[:SOURCE]:RADio:ARB:HEADer:RMS? <"file_name">
```

This command sets the RMS value in the header file for the waveform designated by the <"file\_name"> variable. The RMS value is expressed in volts. The filename variable includes the directory path and can designate a file in either the WFM1, NVWFM, or SEQ directories. For information on the file name syntax, refer to [File Name Variables](#) and [ARB Waveform File Directories](#). When a file is created with no header information then a header is automatically generated with all fields set to unspecified.

The <val> variable is the user-measured RMS value for the specified waveform. The UNSPecified parameter means that the signal generator will calculate the RMS value when it is needed. The signal generator calculation includes rise times and does not include consecutive zero level samples. DC offsets and noise are also included in the RMS measurement. Because the RMS calculation, done by the signal generator, is slow and may not be appropriate for your application it is recommended that the user calculate and enter in their measured RMS value for the waveform file.

The RMS value is calculated as:

$$\sqrt{\sum_{n=1}^N \left( i_n^2 + q_n^2 \right) * \frac{1}{N}}$$

N = # of Samples

### Example

```
[ :SOURCE]:RADio:ARB:HEADer:RMS "WFM1:Sine_Wave",.835  
:RAD:ARB:HEADer:RMS "WFM1:Sine_Wave",UNSP
```

The first example sets a user-measured RMS value for the Sine\_Wave waveform file in the waveform's header file. The second example, the signal generator will calculate the RMS value when needed.

**Range** 0 to 1.414213562373095

## :HEADer:SAVE

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:HEADer:SAVE
```

This command saves the header information to the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

The dual ARB must be On for this command to function and is enabled by the [:STATe] command.

**Key Entry**            **Save Setup To Header**

## :IQ:EXTeRnal:FILTer

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:IQ:EXTeRnal:FILTer 40e6|THROUGH  
[ :SOURce]:RADio:ARB:IQ:EXTeRnal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs. The filter has no effect on the modulated RF signal. Selecting a filter using this command will automatically set the :IQ:EXTeRnal:FILTer:AUTO command to OFF.

40e6                    This choice applies a 40 MHz baseband filter.

THROUGH                This choice selects the through path.

### Example

```
:RAD:ARB:IQ:EXT:FILT 40E6
```

The preceding example selects a 40 MHz filter for the I/Q signals routed to the rear panel.

```
*RST                    THR
```

**Key Entry**            **40.000 MHz**            **Through**

## :IQ:EXTeRnal:FILTer:AUTO

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:IQ:EXTeRnal:FILTer:AUTO ON|OFF|1|0  
[ :SOURce]:RADio:ARB:IQ:EXTeRnal:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals routed to the rear panel I/Q outputs.

ON(1)                    This choice automatically selects the 40 MHz filter optimized for the current signal generator settings.

OFF(0)                    This choice disables the auto feature and allows you to select the 40 MHz filter or a through path. Refer to the :IQ:EXTeRnal:FILTer command for selecting a filter or through path.

### Example

```
:RAD:ARB:IQ:EXT:FILT:AUTO OFF
```

The preceding example disables the automatic filter selection.

**\*RST** 1  
**Key Entry** I/Q Output Filter Manual Auto

### **:IQ:MODulation:ATTen**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:IQ:MODulation:ATTen <val><units>  
[:SOURCE]:RADio:ARB:IQ:MODulation:ATTen?
```

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path. The variable <val> is expressed in decibels (dB)

#### **Example**

```
:RAD:ARB:IQ:MOD:ATT 20
```

The preceding example sets the attenuator level to 20 dB.

**\*RST** +2.00000000E+000  
**Range** 0 to 40 dB  
**Key Entry** Modulator Atten Manual Auto

### **:IQ:MODulation:ATTen:AUTO**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0  
[:SOURCE]:RADio:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the modulator attenuator auto mode. The auto mode will be switched to manual if the signal generator receives an AUTO OFF or AUTO 0 command.

ON (1) This choice sets the modulator attenuator to auto mode which optimizes the attenuation setting for the current signal generator settings.

OFF (0) This choice sets the attenuator to manual mode and holds the attenuator at its current setting. Refer to the [:IQ:MODulation:ATTen](#) command for setting the attenuation value.

#### **Example**

```
:RAD:ARB:IQ:MOD:ATT:AUTO 0
```

The preceding example selects the modulator attenuator manual mode.

**\*RST** 1  
**Key Entry** Modulator Atten Manual Auto

## **:IQ:MODulation:FILTer**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:IQ:MODulation:FILTer 40e6|THROUGH
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. This filter has no effect on the I/Q signal out the rear panel. Selecting a filter using this command will automatically set the **:IQ:MODulation:FILTer:AUTO** command to OFF.

**40E6** This choice applies a 40 MHz baseband filter to the I/Q signals.

**THROUGH** This choice selects the through path.

### **Example**

```
:RAD:ARB:IQ:MOD:FILT 40E6
```

The preceding example selects a 40 MHz filter.

```
*RST THR
```

**Key Entry**      **40.000 MHz**      **Through**

## **:IQ:MODulation:FILTer:AUTO**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals modulated onto the RF carrier.

**ON (1)** This choice will automatically select optimized filters for the current signal generator setting.

**OFF (0)** This choice disables the automatic filter selection. A digital modulation filter or through path may now be selected by using the **:IQ:MODulation:FILTer** command.

### **Example**

```
:RAD:ARB:IQ:MOD:FILT:AUTO 1
```

The preceding example allows for automatic filter selection.

```
*RST 1
```

**Key Entry**      **I/Q Mod Filter Manual Auto**

## :MARKer:CLEar

**Supported** ES267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:MARKer:CLEar "<file_name>", <marker>, <first_point>,  
<last_point>
```

This command clears a single marker point or a range of marker points on a waveform segment for the selected marker (1–4). The Dual ARB mode and all of the ARB modes use this command.

**"<file\_name>"** This variable specifies the name of the waveform file in volatile waveform memory (WFM1). Use the AUTOGEN\_WAVEFORM file when clearing marker points for the currently active ARB format and then save the file using a different file name. The PSG automatically creates a file, using current settings, and names it AUTOGEN\_WAVEFORM whenever an ARB format is turned on (except Dual ARB); the same file name is used for all ARB formats. When all ARB formats are off, this file will still be in waveform memory (WFM1) and is available for use by the Dual ARB. For information on the file name syntax, refer to [“File Name Variables” on page 10](#).

**<marker>** This variable selects the marker number; an integer value from one to four.

**<first\_point>** This variable defines the first point in a range of points. The number must be greater than or equal to one, and less than or equal to the total number of waveform points.

If you enter a value for either the first marker point or the last marker point that would make the first marker point occur after the last, the last marker point automatically adjusts to match the first marker point.

**<last\_point>** This variable defines the last point in a range of points. The number must be greater than or equal to the first point, and less than or equal to the total number of waveform points.

To clear a single marker point, use the same marker point for the first and last point variables. For more information on markers and ARB files, refer to the *User’s Guide*.

### Example

```
:RAD:ARB:MARK:CLE "Test_Data", 1, 1, 300
```

The preceding example clears marker 1 from the first point through the 300th point in the Test\_Data file.

**Range**                   <marker>: 1 to 4  
                          <first\_Point>: 1–number of waveform points  
                          <last\_point>: <first\_Point>–number of waveform points

**Key Entry**           **Set Marker Off Range Of Points**   **Marker 1 2 3 4**   **First Mkr Point**   **Last Mkr Point**

## **:MARKer:CLEar:ALL**

**Supported** E8267D with Option 601 or 602

[[:SOURCE]:RADio:ARB:MARKer:CLEar:ALL "<file\_name>", <marker>

This command clears all marker points on a waveform segment for the selected marker (1–4). The Dual ARB player and all of the ARB formats use this command. With all marker points cleared, the event output signal level is set low.

"<file\_name>" This variable specifies the name of the waveform file in volatile waveform memory (WFM1). Use the AUTOGEN\_WAVEFORM file when clearing all marker points for the currently active ARB format and then save the file using a different file name. The PSG automatically creates a file, using current settings, and names it AUTOGEN\_WAVEFORM whenever an ARB format is turned on (except Dual ARB); the same file name is used for all ARB formats. When all ARB formats are off, this file will still be in waveform memory (WFM1) and is available for use by the Dual ARB. For information on the file name syntax, refer to [“File Name Variables” on page 10](#).

<marker> This variable selects the marker number; an integer value from one to four.

### **Example**

```
:RAD:ARB:MARK:CLE:ALL "Test_Data",1
```

The preceding example clears marker 1 from the all waveform points in the Test\_Data file.

**Range** 1 to 4

**Key Entry** **Marker 1 2 3 4 Set Marker Off All Points**

## **:MARKer:ROtate**

**Supported** E8267D with Option 601 or 602

[[:SOURCE]:RADio:ARB:MARKer:ROtate "<file\_name>", <rotate\_count>

This command shifts the marker points for all markers in a waveform earlier or later by the value of the <rotate\_count> variable. The Dual ARB player and all of the ARB formats use this command.

You can use a positive or negative value. When a marker point is close to the end of the waveform and the <rotate\_count> value is greater than the number of remaining marker points, but less than the total number of marker points, the marker points that would move beyond the end of the waveform wrap to the beginning of the waveform. For example, if a marker point resides at sample point 195 out of 200, and the <rotate\_count> value is twenty-five, the marker point wraps to the beginning of the waveform and continues out to the twentieth waveform point.

To set the marker points in a waveform, refer to [“:MARKer:\[SET\]” on page 256](#).

"<file\_name>" This variable specifies the name of the waveform file in volatile waveform memory (WFM1). Use the AUTOGEN\_WAVEFORM file when rotating marker points for the currently active ARB format and then save the file using a different file name. The PSG automatically creates a file, using current settings, and names it AUTOGEN\_WAVEFORM whenever an ARB format is turned on (except Dual ARB); the same file name is used for all ARB formats. When all ARB formats are off, this file will still be in waveform memory (WFM1) and is available for use by the Dual ARB. For information on the file name syntax, refer to [“File Name Variables”](#)

on page 10.

### Example

```
:RAD:ARB:MARK:ROT "Test_Data",100
```

The preceding example shifts all markers set in the Test\_Data file 100 points later. If the first set point in the file is at 50, then after sending this command, the first set point will be 150 (assuming the Test\_Data file has at least 150 points) and no later set points wrapped around to the beginning of the file.

**Range**                    - (n - 1) to (n - 1)  
                              n = number of points in the waveform

### :MARKer:[SET]

**Supported**                E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:MARKer:[SET] "<file_name>",<marker>,<first_point>,<last_point>,<skip_count>
```

This command sets a single marker point or a range of marker points on a waveform segment for the selected marker (1–4). The Dual ARB player and all of the ARB formats use this command.

The PSG provides four independent markers. Each marker routes an output signal to the rear panel event connector number (BNC—EVENT 1 and EVENT 2 or AUXILIARY I/O—EVENT 3 and EVENT 4) that corresponds to the marker number. A marker consists of marker points placed at defined sample points in a waveform segment. This means that a marker point cannot be less than one or greater than the last sample point in the waveform. Marker points are cumulative, so multiple command executions with different range values, without first clearing the existing points, places additional marker points on the waveform. Because of this cumulative behavior, it is a good practice to clear existing marker points prior to setting new points. This will eliminate unexpected marker pulses. Refer to :MARKer:CLEar and :MARKer:CLEar:ALL commands for information on clearing marker points.

For waveforms generated on the signal generator (baseband generator), the PSG automatically places a marker point at the first waveform sample for markers one and two.

---

**NOTE**    You can set markers for either positive or negative polarity. The following discussions for this command assume positive marker polarity. When using negative marker polarity, the marker pulses occur during the periods of no marker points.

---

There are three ways to place marker points using this command:

- consecutive marker points over a range that collectively create a single marker pulse that spans the range
- equally spaced marker points over a range, so that a marker pulse occurs at each sample point that coincides with a marker point (Using this method, you can configure a clock signal by setting the <skip\_count> variable to one.)
- a single marker point placed at a specific sample point in the waveform, which outputs a single pulse relative to the marker point location (To configure a single marker point, set the first and last points to the same number.)

For more information on markers, refer to the *User's Guide*.



The following list describes the command variables:

"<file_name>"	This variable specifies the name of the waveform file in volatile waveform memory (WFM1). Use the AUTOGEN_WAVEFORM file when setting marker points for the currently active ARB format and then save the file using a different file name. The PSG automatically creates a file, using current settings, and names it AUTOGEN_WAVEFORM whenever an ARB format is turned on (except Dual ARB); the same file name is used for all ARB formats. When all ARB formats are off, this file will still be in waveform memory (WFM1) and is available for use by the Dual ARB. For information on the file name syntax, see “File Name Variables” on page 10
<marker>	This variable selects the marker number; an integer value from one to four.
<first_point>	This variable defines the first point in the range over which the marker is placed. This number must be greater than or equal to one, and less than or equal to the total number of waveform points.  If you enter a value for either the first marker point or the last marker point that would make the first marker point occur after the last, the last marker point is automatically adjusted to match the first marker point.
<last_point>	This variable defines the last point in the range over which the marker will be placed. This value must be greater than or equal to the first point, and less than or equal to the total number of waveform points.
<skip_count>	This variable defines the marker point pattern across the range. A zero value means the marker points occur consecutively across the range. A value greater than zero creates a repeating marker point pattern across the range, where the gap between the marker points is equal to the <skip_count> value. The gaps begin after the first marker point. Each marker point in the pattern, which is only one point wide, produces a marker pulse.

### Example

```
:RAD:ARB:MARK "Test_Data",1,40,100,2
```

The preceding example sets marker 1 on the first point, 40, the last point, 100, and every third point (skip 2) between 40 and 100 (assuming the Test\_Data file has at least 100 points).

<b>Range</b>	<marker>: 1–4 <first_point>: 1–number of waveform points <last_point>: <first_point>–number of waveform points <skip_count>: 0–number of points in the range
<b>Key Entry</b>	<b>Set Marker on Range Of Points    Marker 1 2 3 4    First Mkr Point    Last Mkr Point</b> <b># Skipped Points    Apply to Waveform</b>

## :MDEStination:AAMPLitude

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:MDEStination:AAMPLitude NONE | M1 | M2 | M3 | M4  
[:SOURCE]:RADio:ARB:MDEStination:AAMPLitude?
```

This command routes the selected marker to the Alternate Amplitude function. The NONE parameter clears the marker for the Alternate Amplitude function.

\*RST NONE

<b>Key Entry</b>	<b>None</b>	<b>Marker 1</b>	<b>Marker 2</b>	<b>Marker 3</b>	<b>Marker 4</b>
------------------	-------------	-----------------	-----------------	-----------------	-----------------

## :MDEStination:ALCHold

**Supported** E8267D with Option 601 or 602

---

**CAUTION** Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:ARB:MDEStination:ALCHold NONE | M1 | M2 | M3 | M4  
[:SOURCE]:RADio:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker. For setting markers, see “:MARKer:[SET]” on page 256.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC leveling circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker’s polarity, see “:MPOLarity:MARKer1|2|3|4” on page 260.

---

**NOTE** Do not use the ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings.

---

For more information on the marker ALC hold function, see the *User’s Guide*. For setting the marker points, see “:MARKer:[SET]” on page 256.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
*RST	NONE

### Example

```
:RAD:ARB:MDES:ALCH M1
```

The preceding example routes marker 1 to the ALC Hold function.

**Key Entry**            **None**   **Marker 1**   **Marker 2**   **Marker 3**   **Marker 4**

### :MDEStination:PULSe

**Supported**            E8267D with Option 601 or 602

---

**CAUTION**    The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4
[:SOURCE]:RADio:ARB:MDEStination:PULSe?
```

This command enables or disables the marker pulse/RF blanking function for the selected marker. The function automatically uses the ALC hold function, so there is no need to select both ALC hold and marker pulse/RF blanking functions for the same marker

---

**NOTE**    Do not use ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

---

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker's polarity, refer to the :MPOLarity:MARKer1|2|3|4 command.

---

**NOTE**    Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This causes either no RF output or a continuous RF output. Refer to the :MARKer:[SET] command for setting the marker points.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin. The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking.

---

For more information on the marker RF blanking function, refer to the *User's Guide*.

**NONE** This terminates the marker RF blanking/pulse function.

**M1–M4** These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

**Example**

```
:RAD:ARB:MDES:PULS M2
```

The preceding example routes marker 2 to Pulse/RF Blanking.

```
*RST NONE
```

<b>Key Entry</b>	<b>None</b>	<b>Marker 1</b>	<b>Marker 2</b>	<b>Marker 3</b>	<b>Marker 4</b>
------------------	-------------	-----------------	-----------------	-----------------	-----------------

**:MPOLarity:MARKer1|2|3|4**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive  
[:SOURCE]:RADio:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker. For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points.

**Example**

```
:RAD:ARB:MPOL:MARK3 NEG
```

The preceding example sets the polarity for marker 3 to negative.

```
*RST POS
```

<b>Key Entry</b>	<b>Marker 1 Polarity Neg Pos</b>	<b>Marker 2 Polarity Neg Pos</b>	<b>Marker 3 Polarity Neg Pos</b>
	<b>Marker 4 Polarity Neg Pos</b>		

## :NOISe

**Supported** E8267D with Option 601 or 602 and Option 403

```
[ :SOURce]:RADio:ARB:NOISe[:STATe] ON|OFF|1|0
[:SOURce]:RADio:ARB:NOISe[:STATe]?
```

This command enables or disables adding real-time, non-repeating additive white gaussian noise (AWGN) to the carrier modulated by the waveform being played by the Dual ARB waveform player. The noise bandwidth will be at least 0.8 times the sample rate, or 1.6 times the sample rate depending on the bandwidth factor. For information on the bandwidth factor, refer to the “:NOISe:BFACTOR” command.

When the bandwidth factor is 2, and the sample rate is greater than 50 Msamples/sec, noise cannot be enabled. Maximum bandwidth cannot exceed 80 MHz. Any oversampling in the waveform increases the noise bandwidth by a factor equal to the oversampling.

### Example

```
:RAD:ARB:NOIS ON
```

The preceding example applies real-time AWGN to the carrier.

```
*RST 0
```

**Key Entry** Real-time Noise Off On

## :NOISe:BFACTOR

**Supported** E8267D with Option 601 or 602 and Option 403

```
[ :SOURce]:RADio:ARB:NOISe:BFACTOR <1 - 2>
[:SOURce]:RADio:ARB:NOISe:BFACTOR?
```

This command sets the flat noise bandwidth for applied real time noise. The bandwidth factor will set the noise bandwidth to at least 0.8 times the sample rate when the bandwidth factor is 1 or to 1.6 times the sample rate if the bandwidth factor is 2. Maximum bandwidth cannot exceed 80 MHz.

When the bandwidth factor is 2, and the sample rate is greater than 50 megasamples/sec, noise cannot be enabled. Any oversampling in the waveform increases the noise bandwidth by a factor equal to the oversampling.

### Example

```
:RAD:ARB:NOIS:BFAC 2
```

The preceding example sets the bandwidth factor to 2 and increases the flat noise bandwidth by at least 1.6 times the ARB sample clock rate.

```
*RST +1
```

**Key Entry** Noise Bandwidth Factor

## :NOISe:CBWidth

**Supported** E8267D with Option 601 or 602 and Option 403

```
[ :SOURce]:RADio:ARB:NOISe:CBWidth <1Hz-80MHz>  
[:SOURce]:RADio:ARB:NOISe:CBWidth?
```

This command selects the carrier bandwidth over which the AWGN (additive white gaussian noise) is applied. The noise power will be integrated over the selected bandwidth for the purposes of calculating C/N (carrier to noise ratio). The carrier bandwidth is limited to the ARB sample rate but cannot exceed 80 MHz. For more information refer to the “:NOISe” and :NOISe:BFACTOR commands.

**\*RST** +1.00000000E+000  
1.0 Hz

**Range** 1 Hz to 80 MHz

**Key Entry** **Carrier Bandwidth**

## :NOISe:CN

**Supported** E8267D with Option 601 or 602 and Option 403

```
[ :SOURce]:RADio:ARB:NOISe:CN <-100dB - 100dB>  
[:SOURce]:RADio:ARB:NOISe:CN?
```

This command sets the carrier to noise ratio in dB. The carrier power is defined as the total modulated signal power without noise power added. The noise power is applied over the specified bandwidth of the carrier signal. For more information, refer to the “:NOISe:CBWidth” command.

### Example

```
:RAD:ARB:NOIS:CN 50DB
```

The preceding example sets the carrier to noise ratio to 50 dB.

**\*RST** +0.00000000E+000

**Key Entry** **Carrier to Noise Ratio**

## :NOISe:MUX

**Supported** E8267D with Option 601 or 602 and Option 403

```
[ :SOURce]:RADio[1]|2|3|4:ARB:NOISe:MUX SUM|CARRier|NOISe  
[:SOURce]:RADio[1]|2|3|4:ARB:NOISe:MUX?
```

This command provides diagnostic control of additive noise, such that only noise, only the carrier, or the sum of both the noise and the carrier will be output from the internal baseband generator. The intended purpose of this feature is to allow direct measurement of just the carrier or noise contribution to the total power (assuming that the ALC is off). The system will still behave as if both the noise and the carrier are present on the output when it comes to determining the Auto Modulation Attenuation and the RMS level for RMS Power Search.

The query returns the current setting of the Output Mux.

### Example

```
:RADio:ARB:NOISe:MUX CARR
```

The preceding example sets the additive noise to CARRIER only.

**Default** Carrier+Noise

**Key Entry** Output Mux

### **:REfERENCE:EXtERnal:FREQUency**

**Supported** E8267D with Option 601 or 602

[ :SOURce]:RADio:ARB:REfERENCE:EXtERnal:FREQUency <val>

[ :SOURce]:RADio:ARB:REfERENCE:EXtERnal:FREQUency?

This command allows you to enter the frequency of the external reference.

The variable <val> is expressed in hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear panel connector.

To specify external as the ARB reference frequency you *must* set the ARB reference to EXtERnal by using the :REfERENCE[:SOURce] command.

#### **Example**

```
:RAD:ARB:REF:EXT:FREQ 500KHZ
```

The preceding example sets the external clock frequency reference to 500 kHz.

**\*RST** +1.00000000E+007

**Range** 2.5E5 to 1E8

**Key Entry** Reference Freq

### **:REfERENCE[:SOURce]**

**Supported** E8267D with Option 601 or 602

[ :SOURce]:RADio:ARB:REfERENCE[ :SOURce] INTernal|EXtERnal

[ :SOURce]:RADio:ARB:REfERENCE[ :SOURce]?

This command selects either an internal or external reference for the waveform clock.

If the EXtERnal choice is selected, the external frequency value *must* be entered and the signal must be applied to the BASEBAND GEN REF IN rear panel connector.

Use the :REfERENCE:EXtERnal:FREQUency command to set the external reference frequency.

#### **Example**

```
:RAD:ARB:REF EXT
```

The preceding example sets the ARB reference to external.

**\*RST** INT

**Key Entry** ARB Reference Ext Int

## :RETRigger

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:RETRigger ON|OFF|IMMediate  
[:SOURce]:RADio:ARB:RETRigger?
```

This command selects the signal generator's response to a trigger signal while using the single trigger mode.

When the PSG receives multiple trigger occurrences, when only one is required, it uses the first trigger and ignores the rest. For more information on triggering and to select the single trigger mode, refer to the :TRIGger:TYPE command.

The following list describes the waveform's response to each of the command choices:

ON	The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. If there is a subsequent trigger during playback, the waveform completes its current playback and then plays one more time. If there is no subsequent trigger, the waveform plays once and stops until it receives another trigger.
OFF	The waveform waits for a trigger before play begins and ignores triggers during playback. To restart the waveform, you must send a trigger after the playback completes.
IMMediate	The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. Upon receipt of the subsequent trigger, the waveform immediately resets and begins playing from the beginning of the file. For a waveform sequence, this means to the beginning of the first segment in the sequence.

### Example

```
:RAD:ARB:RETR IMM
```

The preceding example selects the immediate mode for the single mode trigger.

```
*RST ON
```

<b>Key Entry</b>	<b>On</b>	<b>Off</b>	<b>Immediate</b>
------------------	-----------	------------	------------------

## :RSCAling

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:RSCAling <val>  
[:SOURce]:RADio:ARB:RSCAling?
```

This command adjusts the scaling value that is applied to a waveform while it is playing. The variable <val> is expressed as a percentage. Runtime scaling does not alter the waveform data file. For more information about runtime scaling, refer to the *User's Guide*.

### Example

```
:RAD:ARB:RSC 50
```

The preceding example applies a 50% scaling factor to the selected waveform.

```
*RST +7.00000000E+001
```



**Range** 1 to 100 percent  
**Key Entry** **Waveform Runtime Scaling**

### **:SCALing**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:SCALing "<file_name>" ,<val>
```

This command scales the designated "<file\_name>" waveform file while it is being played by the Dual ARB player. The variable <val> is expressed as a percentage. For information on file name syntax, refer to “File Name Variables” on page 10.

Scaling is additive and permanent. You cannot scale up. If you scale a waveform file by 60% and then scale it again to 80% you will scale down the 60% waveform file. For more information about waveform file scaling, refer to the *User’s Guide*.

#### **Example**

```
:RAD:ARB:SCAL "Test_Data" , 50
```

The preceding example applies a 50% scaling factor to the Test\_Data waveform file.

**Range** 1 to 100 percent  
**Key Entry** **Scaling** **Scale Waveform Data**

### **:SCLock:RATE**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:SCLock:RATE <sample_clock_rate>  

[:SOURCE]:RADio:ARB:SCLock:RATE?
```

This command sets the ARB sample clock rate. The sample\_clock\_rate variable can be set from 1 hertz to 100 megahertz.

#### **Example**

```
:RAD:ARB:SCL:RATE 1E6
```

The preceding example sets the ARB sample clock for 1 MHz.

**\*RST** +1.00000000E+008

**Range** 1 to 1.0E8 Hz

**Key Entry** **ARB Sample Clock**

### **:SEQUence**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:SEQUence  

"<file_name>" , "<waveform1>" , <reps> , NONE | M1 | M2 | M3 | M4 | M1M2 | M1M3 | M1M4 | M2M3 | M2M4 | M3M4 | M1M  

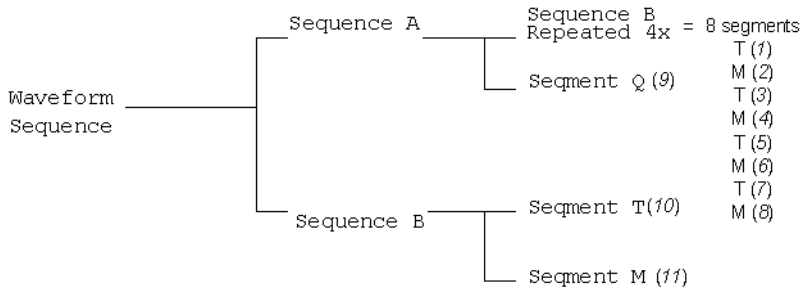
2M3 | M1M2M4 | M1M3M4 | M2M3M4 | ALL , { "<waveform2>" , <reps> , NONE | M1 | M2 | M3 | M4 | M1M2 | M1M3 | M1M4 | M2  

M3 | M2M4 | M3M4 | M1M2M3 | M1M2M4 | M1M3M4 | M2M3M4 | ALL }
```

[ :SOURCE]:RADio:ARB:SEquence? "<file\_name>"

This command creates a waveform sequence. A waveform sequence is made up of segments and other sequences. Any number of segments, up to a segment count limit of 32768, can be used to create a sequence. The count limit is determined by the number of segments in the waveform sequence. Repeated segments are included in the count limit.

For example, using the figure below, suppose a waveform is created using two sequences: Sequence\_A and Sequence\_B. Sequence\_A consists of Sequence\_B and Segment\_Q with Sequence\_B repeated four times. The total segment count for this waveform sequence would be eleven.



The query returns the contents and segment settings of the waveform sequence file

The segments and sequences play in the same order as placed into the waveform sequence by the command. Once you create the file, you cannot edit the segment settings or add further waveform segments unless you use the signal generator's front panel. Using the same waveform sequence name overwrites the existing file with that name. To use a segment's marker settings, you must enable the segment's markers within the segment or within the waveform sequence. A sequence is stored in the catalog of SEQ files USER/SEQ or SEQ: directory.

When you create a waveform sequence, the PSG also creates a file header for the sequence. This file header takes priority over segment or nested sequence file headers. Refer to the *User's Guide* for more information on file headers. To save the file header, see [":HEADer:SAVE" on page 251](#).

- "<file\_name>" This variable names the waveform *sequence* file. For information on the file name syntax, see ["File Name Variables" on page 10](#).
- "<waveform1>" This variable specifies the name of an existing waveform *segment* or sequence file. A waveform segment or the waveform segments in a specified sequence must reside in volatile memory, WFM1, before it can be played by the Dual ARB player. For more information on waveform segments, refer to the *User's Guide*.
- "<waveform2>" This variable specifies the name of a second existing waveform *segment* or sequence file. The same conditions required for waveform1 apply for this segment or sequence. Additional segments and other sequences can be inserted into the file.
- <reps> This variable sets the number of times a segment or sequence plays (repeats) before the next segment or sequence plays.

NONE	This choice disables all four markers for the waveform. Disabling markers means that the waveform sequence ignores the segment's or sequence's marker settings.
M1, M2, M3, M4	These choices, either individually or a combination of them, enable the markers for the waveform segment or sequence. Markers not specified are ignored for that segment or sequence.
ALL	This choice enables all four markers in the waveform segment or sequence.

### Example

```
:RAD:ARB:SEQ"SEQ:Test_Data", "WF1:ramp_test_wfm", 25, M1M4, "WF1:sine_test_wfm", 100, ALL
```

---

**NOTE** A carriage return or line feed is never included in a SCPI command. The example above contains a carriage return so that the text will fit on the page.

---

The preceding example creates a waveform sequence file named Test\_Data. This file consists of the factory supplied waveform segments, ramp\_test\_wfm and sine\_test\_wfm. The waveform is stored in the signal generator's SEQ: directory.

- The first segment, ramp\_test\_wfm, has 25 repetitions with markers 1 and 4 enabled.
- The second segment, sine\_test\_wfm, has 100 repetitions with all four markers enabled.

**Range** <reps>: 1 to 65535

**Key Entry**      **Edit Repetitions**    **Toggle Marker 1**    **Toggle Marker 2**    **Toggle Marker 3**  
                                 **Toggle Marker 4**

### :TRIGger:TYPE

**Supported**      E8267D with Option 601 or 602

```
[[:SOURCE]:RADio:ARB:TRIGger:TYPE CONTInuous|SINGle|GATE|SADVance  
[[:SOURCE]:RADio:ARB:TRIGger:TYPE?
```

This command sets the trigger mode (type) that controls how the waveform plays.

Triggers control the playback by telling the PSG when to transmit the modulating signal (waveform). Depending on the trigger settings for the PSG, the waveform transmission can occur once, continuously, or the PSG may start and stop the transmission repeatedly (GATE mode). For waveform sequences, you can even control when each segment plays (SADVance—segment advance mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state. It is common to have multiple triggers occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the waveform. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the waveform's final I and Q levels determine whether you will see the carrier signal or not (zero = no carrier, other values = carrier visible). At the end of most files, the final I and Q points are set to a value other than zero. If desired, you can create and download an external file with the initial I and Q voltages set to values other than zero. Conversely, you can set the last I and Q points to zero. Refer to the *Agilent Signal Generators Programming Guide*.

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the waveform’s transmission.
- Setting the waveform’s response to triggers:
  - CONTinuous, see “:TRIGger:TYPE:CONTInuous[:TYPE]” on page 269.
  - SINGle, see “:TRIGger:TYPE:CONTInuous[:TYPE]” on page 225.
  - SADVance, see “:TRIGger:TYPE:SADVance[:TYPE]” on page 270.
  - GATE, selecting the mode also sets the response
- Selecting the trigger source using the :TRIGger[:SOURce] command, determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:
  - CONTinuous, SINGle, and SADVance, see “:TRIGger[:SOURce]:EXTernal:SLOPe” on page 273
  - GATE, see “:TRIGger:TYPE:GATE:ACTive” on page 269

For more information on triggering, refer to the *User’s Guide*.

The following list describes the trigger type command choices:

CONTinuous	Upon triggering, the waveform repeats continuously.
SINGle	Upon triggering, the waveform segment or sequence plays once.
GATE	An external trigger signal repeatedly starts and stops the waveform’s playback (transmission). The length of each transmission depends on the duty period of the trigger signal and the gate polarity selection (refer to the :TRIGger:TYPE:GATE:ACTive command). The waveform plays during the inactive state and stops during the active polarity selection state. The active state can be set high or low. The gate mode works only with an external trigger source.

---

**NOTE** The ARB gating behavior described above is opposite to the gating behavior for real-time custom.

---

SADVance	The trigger controls the segment advance within a waveform sequence. To use this choice, a waveform sequence must be the active waveform. Ensure that all segments in the sequence reside in volatile memory.
----------	---

**\*RST** CONT

**Example**

```
:RAD:ARB:TRIG:TYPE GATE
```

The preceding example selects the gated trigger mode.

**Key Entry**            **Continuous**    **Single**    **Gated**    **Segment Advance**

## **:TRIGger:TYPE:CONTInuous[:TYPE]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger:TYPE:CONTInuous[:TYPE] FREE|TRIGger|RESet
[:SOURce]:RADio:ARB:TRIGger:TYPE:CONTInuous[:TYPE]?
```

This command selects the waveform's response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see [“:TRIGger:TYPE” on page 267](#).

The following list describes the waveform's response to each of the command choices:

<b>FREE</b>	Turning the ARB format on immediately triggers the waveform. The waveform repeats until you turn the format off, select another trigger, or choose another waveform file.
<b>TRIGger</b>	The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously until you turn the format off, select another trigger, or choose another waveform file.
<b>RESet</b>	The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously. Subsequent triggers reset the waveform to the beginning. For a waveform sequence, this means to the beginning of the first segment in the sequence.

### **Example**

```
:RAD:ARB:TRIG:TYPE:CONT TRIG
```

The preceding example selects the trigger continuous mode.

```
*RST FREE
```

<b>Key Entry</b>	<b>Free Run</b>	<b>Trigger &amp; Run</b>	<b>Reset &amp; Run</b>
------------------	-----------------	--------------------------	------------------------

## **:TRIGger:TYPE:GATE:ACTive**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive LOW|HIGH
[:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode.

The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. When the active state occurs, the PSG stops the waveform playback at the last played sample point, then restarts the playback at the next sample point when the inactive state occurs. For more information on triggering and to select gating as the trigger mode, see [“:TRIGger:TYPE” on page 267](#).

The following list describes the PSG's gating behavior for the polarity selections:

<b>LOW</b>	The waveform playback stops when the trigger signal goes low (active state) and restarts when the trigger signal goes high (inactive state).
<b>HIGH</b>	The waveform playback stops when the trigger signal goes high (active state) and restarts when the trigger signal goes low (inactive state).

### Example

```
:RAD:ARB:TRIG:TYPE:GATE:ACTIVE HIGH
```

The preceding example sets the active gate state to high.

```
*RST HIGH
```

**Key Entry**            **Gate Active Low High**

### **:TRIGger:TYPE:SADVance[:TYPE]**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE] SINGLE|CONTinuous  
[:SOURCE]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE]?
```

This command selects the waveform's response to a trigger signal while using the segment advance (SADVance) trigger mode.

When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest. For more information on triggering and to select segment advance as the trigger mode, see [“:TRIGger:TYPE” on page 267](#).

The following list describes the waveform's response to each of the command choices:

**SINGLE**                Each segment in the sequence requires a trigger to play, and a segment plays only once, ignoring a segment's repetition value (refer to the [:SEQUence](#) command for repetition information). The following list describes a sequence's playback behavior with this choice:

- After receiving the first trigger, the first segment plays to completion.
- When the waveform receives a trigger after a segment completes, the sequence advances to the next segment and plays that segment to completion.
- When the waveform receives a trigger during play, the current segment plays to completion. Then the sequence advances to the next segment, and it plays to completion.
- When the waveform receives a trigger either during or after the last segment in a sequence plays, the sequence resets and the first segment plays to completion.

**CONTinuous**            Each segment in the sequence requires a trigger to play. After receiving a trigger, a segment plays continuously until the waveform receives another trigger. The following list describes a sequence's playback behavior with this choice:

- After receiving the first trigger, the first segment plays continuously.
- A trigger during the current segment play causes the segment to play to the end of the segment file, then the sequence advances to the next segment, which plays continuously.
- When last segment in the sequence receives a trigger, the sequence resets and the first segment plays continuously.

### Example

```
:RAD:ARB:TRIG:TYPE:SADV CONT
```

The preceding example selects the continuous segment advance mode.

```
*RST          CONT
```

<b>Key Entry</b>	<b>Single</b>	<b>Continuous</b>
------------------	---------------	-------------------

### :TRIGger[:SOURce]

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger[:SOURce] KEY|EXT|BUS  
[:SOURce]:RADio:ARB:TRIGger[:SOURce]?
```

This command sets the trigger source.

For more information on triggering, see “:TRIGger:TYPE” on page 267. The following list describes the command choices:

**KEY** This choice enables manual triggering by pressing the front panel **Trigger** hardkey.

**EXT** An externally applied signal triggers the waveform. This is the only choice that works with gating. The following settings affect an external trigger:

- The input connector for the trigger signal. You have a choice between the rear panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector. To make the connector selection, see “:TRIGger[:SOURce]:EXTernal[:SOURce]” on page 272.

For more information on the connectors and on connecting the cables, refer to the *User’s Guide*.

- The trigger signal polarity:
  - gating mode, see “:TRIGger:TYPE:GATE:ACTive” on page 269
  - continuous, single, and segment advance modes, see “:TRIGger[:SOURce]:EXTernal:SLOPe” on page 273
- The time delay between when the PSG receives a trigger and when the waveform responds to the trigger. There are two parts to setting the delay:
  - setting the amount of delay, see “:TRIGger[SOURce]:EXTernal:DELay” on page 272
  - turning the delay on, see “:TRIGger[:SOURce]:EXTernal:DELay:STATe” on page 273

**BUS** This choice enables triggering over the GPIB using the \*TRG or GET commands, or the LAN and the AUXILIARY INTERFACE (RS–232) using the \*TRG command.

### Example

```
:RAD:ARB:TRIG KEY
```

The preceding example sets the trigger source to manual, front panel key operation.

```
*RST          EXT
```

<b>Key Entry</b>	<b>Trigger Key</b>	<b>Ext</b>	<b>Bus</b>
------------------	--------------------	------------	------------

## **:TRIGger[:SOURce]:EXTErnal[:SOURce]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTErnal[:SOURce] EPT1|EPT2|  
EPTRIGGER1|EPTRIGGER2  
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTErnal[:SOURce]?
```

This command selects which PATTERN TRIG IN connection the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, refer to the **:TRIGger[:SOURce]** command. For more information on the rear panel connectors, refer to the *User's Guide*.

The following list describes the command choices:

EPT1	This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear panel connector.
EPT2	This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.
EPTRIGGER1	This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear panel connector.
EPTRIGGER2	This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.

### **Example**

```
:RAD:ARB:TRIG:EXT EPT2
```

The preceding example sets the trigger source to the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.

```
*RST EPT1
```

<b>Key Entry</b>	<b>Patt Trig In 1</b>	<b>Patt Trig In 2</b>
------------------	-----------------------	-----------------------

## **:TRIGger[SOURce]:EXTErnal:DELAy**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTErnal:DELAy <val>  
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTErnal:DELAy?
```

This command sets the amount of time to delay the PSG's response to an external trigger.

The delay is a path (time) delay between when the PSG receives the trigger and when it responds to the trigger. For example, configuring a trigger delay of two seconds, causes the PSG to wait two seconds after receipt of the trigger before the PSG responds and transmits the waveform.

The delay does not occur until you enable it by the **:TRIGger[:SOURce]:EXTErnal:DELAy:STATE** command. You can set the delay value either before or after turning it on.

For more information on configuring an external trigger source and to select external as the trigger source, see **“:TRIGger[:SOURce]” on page 271**.

The unit of measurement for the variable <val> is in seconds (nanoseconds to seconds).



### Example

```
:RAD:ARB:TRIG:EXT:DEL .2
```

The preceding example sets the external delay to 200 milliseconds.

```
*RST +1.00000000E-003
```

**Range** 1E-8 to 4E1

**Key Entry** Ext Delay Time

### :TRIGger[:SOURce]:EXTernal:DELay:STATe

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay:STATe ON|OFF|1|0|TIME|SAMPles  
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay:STATe?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, use the [:TRIGger\[SOURce\]:EXTernal:DELay](#) command. For more information on configuring an external source, refer to [“:TRIGger\[:SOURce\]” on page 271](#).

ON (1) This choice enables the external delay mode.

OFF (0) This choice disables the external delay mode.

TIME This choice sets the number of seconds delay for the external trigger.

SAMPles This choice sets the delay applied to the external trigger to be in units of I/Q samples.

### Example

```
:RAD:ARB:TRIG:EXT:DEL:STAT OFF
```

The preceding example disables the external delay function.

```
*RST 0, 10 msec, 10 msec
```

**Range**

<i>Parameter</i>	<i>Minimum</i>	<i>Maximum</i>
Time	10 nsec	40 seconds
Samples	10 nsec	40 seconds

**Key Entry** Ext Delay Off On Ext Delay Time Ext Delay Samples

### :TRIGger[:SOURce]:EXTernal:SLOPe

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:SLOPe POSitive|NEGative  
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous, single, or segment advance triggering modes. To set the polarity for gating, see [“:TRIGger:TYPE:GATE:ACTIVE” on page 269](#).

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (plays) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, see “:TRIGger[:SOURce]” on page 271.

### Example

```
:RAD:ARB:TRIG:EXT:SLOP NEG
```

The preceding example sets the external trigger slope to negative.

```
*RST          NEG
```

**Key Entry**           **Ext Polarity Neg Pos**

### :VCO:CLOCK

**Supported**           E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:VCO:CLOCK INTernal|EXTernal  
[:SOURce]:RADio:ARB:VCO:CLOCK?
```

This command selects an internal or external VCO clock. The external VCO clock is connected to the rear panel BASEBAND GEN CLK IN connector. Use the :DACS:ALIGn command after an external VCO clock is first applied to the BASEBAND GEN CLK IN connector or when the VCO signal is lost and then re-acquired.

### Example

```
:RAD:ARB:VCO:CLOC EXT
```

The preceding example selects an external VCO clock.

```
*RST          Int
```

**Key Entry**           **VCO Clock Ext Int**

### :WAVeform

**Supported**           E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:WAVeform "WFM1:file_name" | "SEQ:filename"  
[:SOURce]:RADio:ARB:WAVeform?
```

This command, for the Dual ARB mode, selects a waveform file or sequence, for the Dual ARB player to play. The file must be present in volatile memory, WFM1: or in the SEQ directory. If a file is in non-volatile memory (NVWFM), use the :COPY command to copy the file to WFM1.

"WFM1:file\_name" This variable names a waveform file residing in volatile memory:WFM1. For information on the file name syntax, see “File Name Variables” on page 10.

"SEQ:filename" This variable names a sequence file residing in the catalog of sequence files. For more information on the file name syntax, see “File Name Variables” on page 10.

### Example

```
:RAD:ARB:WAV "WFM1:Test_Data"
```

The preceding example selects the file Test\_Data from the list of files in volatile waveform memory, WFM1 and applies its header settings.

**Key Entry**                    **Select Waveform**

### **:Waveform:NHEAders**

**Supported**                    E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB:WAVEform:NHEAders "WFM1:file_name" | "SEQ:filename"
[:SOURce]:RADio:ARB:WAVEform:NHEAders?
```

This command, for the Dual ARB mode, allows for a fast selection of a waveform file or sequence. No header information or settings are applied to the waveform or sequence when this command is used. This will improve the access or loading speed of the waveform file or sequence to approximately 100 mS for a single segment. The file must be in volatile waveform memory, WFM1: or in the SEQ directory. If a file is in non-volatile memory (NVWFM), use the :COPY command to copy files to WFM1.

"WFM1:file\_name" This variable names a waveform file residing in volatile memory:WFM1. For information on the file name syntax, see [“File Name Variables” on page 10](#).

"SEQ:filename" This variable names a sequence file residing in the catalog of sequence files. For more information on the file name syntax, see [“File Name Variables” on page 10](#).

#### **Example**

```
:RAD:ARB:WAV:NHEA "Test_Data"
```

The preceding example selects the file Test\_Data, without applying header settings.

### **[:STATe]**

**Supported**                    E8267D with Option 601 or 602

```
[ :SOURce]:RADio:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:ARB[:STATe]?
```

This command enables or disables the operating state of the signal generator’s dual arbitrary waveform (ARB) generator.

#### **Example**

```
:RAD:ARB 1
```

The preceding example turns on the signal generator’s ARB generator personality.

```
*RST                            0
```

**Key Entry**                    **ARB Off On**

## Dmodulation Subsystem—Option 601 or 602 ([:SOURCE]:RADIO:DMODulation:ARB)

### :IQ:EXtErnal:FiLTer

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:IQ:EXtErnal:FiLTer 40e6|THRough  
[:SOURCE]:RADIO:DMODulation:ARB:IQ:EXtErnal:FiLTer?
```

This command selects a 40 MHz filter or a through path for I/Q signals routed to the rear panel I and Q outputs. Selecting a filter using this command will automatically set the [:IQ:EXtErnal:FiLTer:AUTO](#) command to OFF.

40e6 This choice selects the 40 MHz baseband filter.

THRough This choice selects a through path and bypasses filtering.

#### Example

```
:RAD:DMOD:ARB:IQ:EXT:FILT 40E6
```

The preceding example selects a 40 MHz filter.

```
*RST THR
```

**Key Entry**      **40.000 MHz**      **Through**

### :IQ:EXtErnal:FiLTer:AUTO

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:IQ:EXtErnal:FiLTer:AUTO ON|OFF|1|0  
[:SOURCE]:RADIO:DMODulation:ARB:IQ:EXtErnal:FiLTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals routed to the rear panel I/Q outputs.

ON(1) This choice automatically selects a filter that is optimized for the current signal generator settings.

OFF(0) This choice disables the auto feature. Select the 40 MHz filter or a through path by using the [:IQ:EXtErnal:FiLTer](#) command.

#### Example

```
:RAD:DMOD:ARB:IQ:EXT:FILT:AUTO 0
```

The preceding example disables the auto mode filter selection.

```
*RST 1
```

**Key Entry**      **I/Q Output Filter Manual Auto**

## :FILTer

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:FILTer RNYQuist|NYQuist|GAUSSian|
RECTangle|AC4Fm|UGGaussian|"<user_FIR>"
[:SOURCE]:RADIO:DMODulation:ARB:FILTer?
```

This command specifies the pre-modulation filter type.

RNYQuist	This choice selects a Root Nyquist (root raised cosine) filter. This filter is adjusted using Alpha.
NYQuist	This choice selects a Nyquist (raised cosine) filter. This filter is adjusted using Alpha.
GAUSSian	This choice selects a Gaussian Filter which is adjusted using Bbt values.
RECTangle	This choice selects a one symbol wide rectangular filter.
AC4Fm	This choice selects a pre-defined Association of Public Safety Communications officials (APCO) specified compatible 4-level frequency modulation (C4FM) filter.
UGGAUSSian	This choice selects a UN3/4 delay-compatible, GSM, 0.300 Bbt Gaussian filter. The Bbt value is not adjustable.
"<User_FIR>"	This variable is any filter file that you have stored in memory. For information on the file name syntax, see <a href="#">“File Name Variables” on page 10</a> .

### Example

```
:RAD:DMOD:ARB:FILTer "FIR:FIR_Data"
```

The preceding example selects a file named FIR\_Data, from the catalog of FIR files, as the filter type.

```
*RST RNYQuist
Key Entry Root Nyquist Nyquist Gaussian Rectangle APCO 25 C4FM
UN3/4 GSM Gaussian User FIR
```

## :FILTer:ALPHa

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:FILTer:ALPHa <val>
[:SOURCE]:RADIO:DMODulation:ARB:FILTer:ALPHa?
```

This command changes the Nyquist or root Nyquist filter alpha value.

The filter alpha value can be set to the minimum level (0), the maximum level (1), or in between by using numeric values (0.001 to 0.999).

To change the current filter type, refer to [“:FILTer” on page 277](#).

### Example

```
:RAD:DMOD:ARB:FILTer:ALPHa .33
```

The preceding example sets .33 as the filter alpha.

```
*RST +3.50000000E-001
```

**Range** 0.000 to 1.000

**Key Entry** **Filter Alpha**

### **:FILTer:BBT**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:FILTer:BBT <val>  
[:SOURCE]:RADio:DMODulation:ARB:FILTer:BBT?
```

This command changes the bandwidth-multiplied-by-bit-time (BbT) filter parameter for a Gaussian filter. It has no effect on other types of filters.

The filter BbT value can be set to the minimum level (0), the maximum level (1), or in between by using fractional numeric values (0.001 to 0.999).

To change the current filter type, refer to “:FILTer” on page 277.

#### **Example**

```
:RAD:DMOD:ARB:FILT:BBT .52
```

The preceding example sets .52 as the filter BbT.

**\*RST** +5.00000000E-001

**Range** 0.000 to 1.000

**Key Entry** **Filter BbT**

### **:FILTer:CHANnel**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:FILTer:CHANnel EVM|ACP  
[:SOURCE]:RADio:DMODulation:ARB:FILTer:CHANnel?
```

This command optimizes the Nyquist and root Nyquist filters to minimize error vector magnitude (EVM) or to minimize adjacent channel power (ACP). To change the current filter type, refer to “:FILTer” on page 277.

#### **Example**

```
:RAD:DMOD:ARB:FILT:CHAN ACP
```

The preceding example selects ACP optimization.

**EVM** This choice provides the most ideal passband.

**ACP** This choice improves stopband rejection.

**\*RST** EVM

**Key Entry** **Optimize FIR For EVM ACP**

## **:HEADer:CLEAr**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ] : RADIO : DMODulation : ARB : HEADer : CLEAr

This command clears the header information from the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the Arb Waveform Generator's Digital Modulation *must* be enabled ON by using the [ :STATe ] command.

**Key Entry**            **Clear Header**

## **:HEADer:SAVE**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ] : RADIO : DMODulation : ARB : HEADer : SAVE

This command saves the header information to the header file for the active modulation file. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the Arb Waveform Generator's Digital Modulation must be enabled ON by using the [ :STATe ] command.

**Key Entry**            **Save Setup To Header**

## **:IQ:MODulation:ATTen**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ] : RADIO : DMODulation : ARB : IQ : MODulation : ATTen <val><unit>  
 [ :SOURCE ] : RADIO : DMODulation : ARB : IQ : MODulation : ATTen?

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path. The variable <val> is expressed in decibels (dB).

### **Example**

```
:RAD:DMOD:ARB:IQ:MOD:ATT 20
```

The preceding example sets the modulator attenuator level to 20 dB.

**\*RST**                    +2.00000000E+000

**Range**                    0 to 40 dB

**Key Entry**            **Modulator Atten Manual Auto**

## **:IQ:MODulation:ATTen:AUTO**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ] : RADIO : DMODulation : ARB : IQ : MODulation : ATTen : AUTO ON | OFF | 1 | 0  
 [ :SOURCE ] : RADIO : DMODulation : ARB : IQ : MODulation : ATTen : AUTO?

This command enables or disables the modulator attenuator auto mode. The auto mode will be switched to manual if the signal generator receives a AUTO OFF or AUTO 0 command.

- ON (1) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.
- OFF (0) This choice holds the attenuator at its current setting or at a selected value set by the [:IQ:MODulation:ATTen](#) command.

### Example

```
:RAD:DMOD:ARB:IQ:MOD:ATT:AUTO ON
```

The preceding example selects the modulator attenuator auto mode.

```
*RST 1
```

**Key Entry**            **Modulator Atten Manual Auto**

### **:IQ:MODulation:FILTer**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer 40e6|THROUGH  
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set the [:IQ:MODulation:FILTer:AUTO](#) command to OFF.

- 40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.
- THROUGH This choice bypasses filtering.

### Example

```
:RAD:DMOD:ARB:IQ:MOD:FILT THR
```

The preceding example selects the through path and bypasses filtering.

```
*RST THR
```

**Key Entry**            **40.000MHz**        **Through**

### **:IQ:MODulation:FILTer:AUTO**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0  
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals modulated onto the RF carrier.

- ON (1) This choice will automatically select a filter that is optimized for the current signal generator setting.
- OFF (0) This choice disables the automatic filter selection. You may select a digital modulation filter or through path by using the [:IQ:MODulation:FILTer](#) command.



**Example**

```
:RAD:DMOD:ARB:IQ:MOD:FILT:AUTO ON
```

The preceding example sets the automatic filter selection function.

```
*RST 1
```

**Key Entry**            **I/Q Mod Filter Manual Auto**

**:MDEStination:ALCHold**

**Supported**            E8267D with Option 601 or 602

---

**CAUTION**    Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:DMODulation:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4  

[:SOURCE]:RADio:DMODulation:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that uses idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see [“:MPOLarity:MARKer1|2|3|4” on page 284](#). For more information on markers, see [“:MARKer:\[SET\]” on page 256](#).

---

**NOTE**    Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE**    A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

---

For more information on the marker ALC hold function, see the *User's Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see [“:MARKer:CLEar” on page 254](#).
- For clearing all marker points, see [“:MARKer:CLEar:ALL” on page 255](#).

- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For setting marker points, see “:MARKer:[SET]” on page 256.

NONE This terminates the marker ALC hold function.  
M1–M4 These are the marker choices. The ALC hold feature uses only one marker at a time.

### Example

```
:RAD:DMOD:ARB:MDES:ALCH M1
```

The preceding example routes marker 1 to the ALC Hold function.

```
*RST NONE
```

**Key Entry**            **None**   **Marker 1**   **Marker 2**   **Marker 3**   **Marker 4**

### :MDEStination:PULSe

**Supported**            E8267D with Option 601 or 602

---

**CAUTION**    The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:DMODulation:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4  
[:SOURCE]:RADio:DMODulation:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

---

**NOTE**    Do not use ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

---

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker’s polarity, refer to the :MPOLarity:MARKer1|2|3|4 command. For more information on markers, see “:MARKer:[SET]” on page 256.

---

**NOTE**    Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

---

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLEar” on page 254.
- For clearing all marker points, see “:MARKer:CLEar:ALL” on page 255.

- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For setting marker points, see “:MARKer:[SET]” on page 256.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

---

For more information on the marker RF blanking function, refer to the *User’s Guide*.

**NONE** This terminates the marker RF blanking/pulse function.

**M1–M4** These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

**Example**

```
:RAD:DMOD:ARB:MDES:PULS M2
```

The preceding example routes marker 2 to the Pulse/RF Blanking function.

```
*RST NONE
```

<b>Key Entry</b>	<b>None</b>	<b>Marker 1</b>	<b>Marker 2</b>	<b>Marker 3</b>	<b>Marker 4</b>
------------------	-------------	-----------------	-----------------	-----------------	-----------------

**:MODulation:FSK[:DEVIation]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:MODulation:FSK[:DEVIation] <val><units>  

[:SOURCE]:RADio:DMODulation:ARB:MODulation:FSK[:DEVIation]?
```

This command sets the symmetric FSK frequency deviation value.

The variable <val> is a numeric expression with a maximum range equal to the current symbol rate value multiplied by ten, limited to 20 MHz. The variable <units> is expressed in hertz.

To change the modulation type use the :MODulation[:TYPE] command. For a list of the minimum and maximum symbol rate values use the :SRATE command.

For more information on setting an asymmetric FSK deviation value, refer to the *User’s Guide*.

**Example**

```
:RAD:DMOD:ARB:MOD:FSK 50KHZ
```

The preceding example sets the maximum frequency deviation to 50 kHz.

```
*RST +4.00000000E+002
```

**Range** 0 to 2E7

**Key Entry** **Freq Dev**

## :MODulation[:TYPE]

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE ] : RADio : DMODulation : ARB : MODulation [ : TYPE ] BPSK | QPSK | IS95QPSK |
GRAYQPSK | OQPSK | IS95OQPSK | P4DQPSK | PSK8 | PSK16 | D8PSK | EDGE | MSK | FSK2 | FSK4 |
FSK8 | FSK16 | C4FM | QAM4 | QAM16 | QAM32 | QAM64 | QAM128 | QAM256
[ : SOURCE ] : RADio : DMODulation : ARB : MODulation [ : TYPE ] ?
```

This command sets the modulation type for the digital modulation personality.

### Example

```
:RAD:DMOD:ARB:MOD BPSK
```

The preceding example selects binary phase shift keying (BPSK) as the modulation type.

<b>*RST</b>	P4DQPSK							
<b>Key Entry</b>	<b>BPSK</b>	<b>QPSK</b>	<b>IS-95 QPSK</b>	<b>Gray Coded QPSK</b>		<b>OQPSK</b>		
	<b>IS-95 OQPSK</b>	$\pi/4$ <b>DQPSK</b>	<b>8PSK</b>	<b>16PSK</b>	<b>D8PSK</b>	<b>EDGE</b>	<b>MSK</b>	
	<b>2-Lvl FSK</b>	<b>4-Lvl FSK</b>	<b>8-Lvl FSK</b>	<b>16-Lvl FSK</b>	<b>C4FM</b>	<b>4QAM</b>	<b>16QAM</b>	
	<b>32QAM</b>	<b>64QAM</b>	<b>128QAM</b>	<b>256QAM</b>	<b>User I/Q</b>	<b>User FSK</b>		

## :MPOLarity:MARKer1|2|3|4

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE ] : RADio : DMODulation : ARB : MPOLarity : MARKer1 | 2 | 3 | 4 NEGative |
POSitive
[ :SOURCE ] : RADio : DMODulation : ARB : MPOLarity : MARKer1 | 2 | 3 | 4 ?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “[:MARKer:CLEar](#)” on page 254.
- For clearing all marker points, see “[:MARKer:CLEar:ALL](#)” on page 255.
- For shifting marker points, see “[:MARKer:ROTate](#)” on page 255.
- For information on markers and setting marker points, see “[:MARKer:\[SET\]](#)” on page 256.

### Example

```
:RAD:DMOD:ARB:MPOL:MARK2 NEG
```

The preceding example sets the polarity for marker 2 to negative.

<b>*RST</b>	POS		
<b>Key Entry</b>	<b>Marker 1 Polarity Neg Pos</b>	<b>Marker 2 Polarity Neg Pos</b>	<b>Marker 3 Polarity Neg Pos</b>
	<b>Marker 4 Polarity Neg Pos</b>		

## **:REFerence:EXTernal:FREQuency**

**Supported** E8267D with Option 601 or 602

```
[[:SOURCE]:RADio:DMODulation:ARB:REFerence:EXTernal:FREQuency <val>  
[:SOURCE]:RADio:DMODulation:ARB:REFerence:EXTernal:FREQuency?
```

This command sets or retrieves the reference frequency value of an externally applied reference to the signal generator. The variable <val> is expressed in hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear panel connector.

To specify EXTernal as the ARB reference source type, use the **:REFerence[:SOURCE]** command.

### **Example**

```
:RAD:DMOD:ARB:REF:EXT:FREQ 10MHZ
```

The preceding example sets the external reference to 10 MHz.

```
*RST +1.00000000E+007
```

**Range** 2.5E5 to 1E8

**Key Entry** Reference Freq

## **:REFerence[:SOURCE]**

**Supported** E8267D with Option 601 or 602

```
[[:SOURCE]:RADio:DMODulation:ARB:REFerence[:SOURCE] INTernal|EXTernal  
[:SOURCE]:RADio:DMODulation:ARB:REFerence[:SOURCE]?
```

This command selects either an internal or external reference for the waveform clock.

If the EXTernal choice is selected, the external frequency value *must* be entered and the signal must be applied to the BASEBAND GEN REF IN rear panel connector.

To enter the external reference frequency use the **:REFerence:EXTernal:FREQuency** command.

### **Example**

```
:RAD:DMOD:ARB:REF INT
```

The preceding example sets an internal clock reference.

```
*RST INT
```

**Key Entry** ARB Reference Ext Int

## :RETRigger

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:RETRigger ON|OFF|IMMediate  
[:SOURce]:RADio:DMODulation:ARB:RETRigger?
```

This command selects the waveform's response to a trigger signal while using the single trigger mode.

When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest. For more information on triggering and to select the single trigger mode, see “:TRIGger:TYPE” on page 291.

The following list describes the waveform's response to each of the command choices:

ON	The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. If there is a subsequent trigger during playback, the waveform completes its current playback and then plays one more time. If there is no subsequent trigger, the waveform plays once and stops until it receives another trigger.
OFF	The waveform waits for a trigger before play begins and ignores triggers during playback. To restart the waveform, you must send a trigger after the playback completes.
IMMediate	The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. Upon receipt of the subsequent trigger, the waveform immediately resets and begins playing from the beginning of the file. For a waveform sequence, this means to the beginning of the first segment in the sequence.

### Example

```
:RAD:DMOD:ARB:RETR ON
```

The preceding example selects the ON mode for the single mode trigger.

```
*RST ON
```

**Key Entry**      **On**    **Off**    **Immediate**

## :SCLock:RATE

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:SCLock:RATE <sample_clock_rate>  
[:SOURce]:RADio:DMODulation:ARB:SCLock:RATE?
```

This command sets the sample clock rate in hertz. The modulation format should be active before executing this command. If this command is executed before the modulation format has been activated by the [:STATe] command, the entered value will be overridden by a calculated factory default value.

**Example**

```
:RAD:DMOD:ARB:SCL:RATE 50E6
```

The preceding example sets the sample clock rate to 50 MHz.

```
*RST          +1.00000000E+008
Range         1 to 1E8
Key Entry     ARB Sample Clock
```

**:SETup**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODULATION:ARB:SETup GSM|NADC|PDC|PHS|DECT|AC4Fm|
ACQPsk|CDPD|PWT|EDGE|TETRA|MCARrier| "<file_name>"
[:SOURCE]:RADIO:DMODULATION:ARB:SETup?
```

This command selects the digital modulation format type. For information on the file name syntax, see [“File Name Variables” on page 10](#).

**Example**

```
:RAD:DMOD:ARB:SET CDPD
```

The preceding example selects cellular digital packet data (CDPD) as the modulation format.

```
*RST          NADC
Key Entry     GSM   NADC   PDC   PHS   DECT   APCO 25 w/C4FM   APCO w/CQPSK
              CDPD   PWT   EDGE  TETRA  Multicarrier Off On   Select File
```

**:SETup:MCARrier**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODULATION:ARB:SETup:MCARrier (GSM|NADC|PDC|PHS|DECT|
AC4Fm|ACQPsk|CDPD|PWT|EDGE|TETRA,<num_carriers>,<freq_spacing>)|
"<file_name>"
[:SOURCE]:RADIO:DMODULATION:ARB:SETup:MCARrier?
```

This command builds a table with the specified number of carriers and frequency spacing or retrieves the setup stored in the specified user file. The query returns the carrier type, number of carriers, and frequency spacing. The output format is as follows:

```
<carrier_type>,<num_carriers>,<freq_spacing>
```

If a specific file is loaded and then queried, only the file name is returned. For information on the file name syntax, see [“File Name Variables” on page 10](#). To store a multicarrier setup refer to [“:SETup:MCARrier:STORE” on page 288](#).

The variable <freq\_spacing> is expressed in hertz (kHz–MHz).

### Example

```
:RAD:DMOD:ARB:SET:MCAR NADC, 2, 10MHZ  
:RAD:DMOD:ARB:SET:MCAR "<file_name>"
```

The preceding examples show the syntax used to select a North American Digital Cellular (NADC) modulation format with two carriers and 10 MHz frequency spacing and the syntax for selecting an existing multicarrier file.

```
*RST          Carrier:      NADC  
              <num carriers>:2  
              <freq spacing>: +1.0000000000000E+06
```

**Range**                   <num carriers>: 2 to 100  
                          <freq spacing>:  $2 \div (\text{<num carriers>} - 1) \times 80$  MHz

<b>Key Entry</b>	<b>GSM</b>	<b>NADC</b>	<b>PDC</b>	<b>PHS</b>	<b>DECT</b>	<b>APCO 25 w/C4FM</b>	<b>APCO w/CQPSK</b>
	<b>CDPD</b>	<b>PWT</b>	<b>EDGE</b>	<b>TETRA</b>	<b># of Carriers</b>	<b>Freq Spacing</b>	
	<b>Custom Digital Mod State</b>						

### :SETup:MCARrier:PHASe

**Supported**           E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:SETup:MCARrier:PHASe FIXed |RANDOM  
[:SOURCE]:RADIO:DMODulation:ARB:SETup:MCARrier:PHASe?
```

This command sets the phase difference between carriers for multicarrier digital modulation.

**FIXed**                This choice sets the phase of all carriers to 0.

**RANDom**              This choice sets random phase values for all of the carriers.

### Example

```
:RAD:DMOD:ARB:SET:MCAR:PHAS RAND
```

The preceding example sets the phase difference between carriers to a random value.

```
*RST          FIX
```

**Key Entry**           **Carrier Phases Fixed Random**

### :SETup:MCARrier:STORe

**Supported**           E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:SETup:MCARrier:STORe "<file_name>"
```

This command stores the current multicarrier setup information.

The stored file contains information that includes the digital modulation format, number of carriers, frequency spacing, and power settings for the multicarrier setup.

The setting enabled by this command is not affected by signal generator power-on, preset, or \*RST. For information on the file name syntax, see [“File Name Variables” on page 10](#).



**Example**

```
:RAD:DMOD:ARB:SET:MCAR:STOR "NADC_Data"
```

The preceding example saves the multicarrier setup information to a file called NADC\_Data and stores the file in the catalog of MDMOD files.

**Key Entry**                    **Load/Store**

**:SETup:MCARrier:TABLE**

**Supported**                    E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:DMODulation:ARB:SETup:MCARrier:TABLE INIT|APPend
<carrier_num>,GSM|NADC|PDC|PHS|DECT|AC4Fm|ACQPsk|CDPD|PWT|EDGE|TETRA|
"<file_name>",<freq_offset>,<power>
[:SOURCE]:RADIO:DMODulation:ARB:SETup:MCARrier:TABLE? <carrier_num>
```

This command modifies the parameters of one of the available multicarrier digital modulation formats.

The variable <freq\_offset> is expressed in units of hertz (kHz to MHz).

The variable <power> is expressed in units of decibels (dB).

The carrier type, carrier name, frequency offset, and power level are returned when a query is initiated. The output format is as follows:

```
<carrier_type>,<carrier_name>,<freq_offset>,<power>
```

**INIT**                            This choice clears the current information and creates a new one row table, allowing for further definition using additional parameters.

**APPend**                         This choice adds rows to an existing table.

**<carrier\_num>**                 This variable specifies the number of the carriers in the multicarrier table that will be modified. The value of the variable <carrier\_num> must be specified prior to selecting the digital modulation format.

For information on the file name syntax, see [“File Name Variables” on page 10](#). To store a multicarrier setup refer to [“.SETup:MCARrier:STORe” on page 288](#).

When a query is initiated, carrier type, frequency offset, and power level are returned in the following format: <carrier\_type>,<freq\_offset>,<power>

```
*RST                            carrier type: NADC
                                 <freq_offset>: 5.00000000E+004
                                 <power>: +0.00000000E+000
```

```
Range                            <freq_offset>: -1E5 to 1E6
                                 <power>: -40 to 0
```

<b>Key Entry</b>	<b>Initialize Table</b>	<b>Insert Row</b>	<b>GSM</b>	<b>NADC</b>	<b>PDC</b>	<b>PHS</b>	<b>DECT</b>
	<b>APCO 25 w/C4FM</b>	<b>APCO w/CQPSK</b>		<b>CDPD</b>	<b>PWT</b>	<b>EDGE</b>	<b>TETRA</b>
	<b>Custom Digital Mod State</b>						

### **:SETup:MCARrier:TABLE:NCARriers**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ]:RADio:DMODulation:ARB:SETup:MCARrier:TABLE:NCARriers?

This query returns the number of carriers in the current multicarrier setup.

**\*RST** +2

**Range** 1 to 100

**Key Entry** # of Carriers

### **:SETup:STORe**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ]:RADio:DMODulation:ARB:SETup:STORe "<file\_name>"

This command stores the current custom digital modulation state using the "<file\_name>" file name.

The saved file contains information that includes the modulation type, filter and symbol rate for the custom modulation setup.

For information on the file name syntax, see [“File Name Variables” on page 10](#).

#### **Example**

```
:RAD:DMOD:ARB:SET:STOR "Setup_Data"
```

The preceding example saves the modulation format setup to a file named Setup\_Data and stores the file in the catalog of DMOD files.

**Key Entry** Store Custom Dig Mod State

### **:SRATe**

**Supported** E8267D with Option 601 or 602

[ :SOURCE ]:RADio:DMODulation:ARB:SRATe <val>

[ :SOURCE ]:RADio:DMODulation:ARB:SRATe?

This command sets the transmission symbol rate. The variable <val> is expressed in symbols per second (sps–Msps) and the maximum range value is dependent upon the source of data (internal or external), the modulation type, and filter.

When user-defined filters are selected using the [:FILTeR](#) command, the upper bit rate will be restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Msps
- FIR filter length > 16 symbols: upper limit is 25 Msps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated as follows:

- Above 12.5 Msps, the FIR length is truncated to 32 symbols
- Above 25 Msps, the FIR length is truncated to 16 symbols

This impacts the relative timing of the modulated data, as well as the actual filter response.

The modulation type is changed using the `:MODulation[:TYPE]` command.

### Example

```
:RAD:DMOD:ARB:SRAT 10KSPS
```

The preceding example sets the symbol rate to 10K symbols per second.

**\*RST**                    +2.43000000E+004

**Range**                    1 kspS to 50 Msps

**Key Entry**                **Symbol Rate**

### :TRIGger:TYPE

**Supported**                E8267D with Option 601 or 602

```
[[:SOURCE]:RADIO:DMODulation:ARB:TRIGger:TYPE CONTInuous|SINGle|GATE  
[:SOURCE]:RADIO:DMODulation:ARB:TRIGger:TYPE?
```

This command sets the trigger mode (type) that controls the waveform's playback.

Triggers control the playback by telling the PSG when to play the modulating signal (waveform). Depending on the trigger settings for the PSG, the waveform playback can occur once, continuously, or the PSG may start and stop playing the waveform repeatedly (GATE mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state of the trigger signal. It is common to have multiple triggers, also referred to as trigger occurrences or events, occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the waveform. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the waveform's final I and Q levels determine whether you will see the carrier signal or not (zero = no carrier, other values = carrier visible). At the end of most files, the final I and Q points are set to a value other than zero.

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the waveform's transmission.
- Setting the waveform's response to triggers:
  - CONTInuous, see “:TRIGger:TYPE:CONTInuous[:TYPE]” on page 292
  - SINGle, see “:RETRigger” on page 286
  - GATE, selecting the mode also sets the response

- Selecting the trigger source using the `:TRIGger[:SOURCE]` command, determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:
  - CONTInuous and SINGle see “`:TRIGger[:SOURCE]:EXTernal:SLOPe`” on page 296
  - GATE, see “`:TRIGger:TYPE:GATE:ACTive`” on page 293

For more information on triggering, refer to the *User’s Guide*.

The following list describes the trigger type command choices:

CONTInuous	Upon triggering, the waveform repeats continuously.
SINGle	Upon triggering, the waveform segment or sequence plays once.
GATE	An external trigger signal repeatedly starts and stops the waveform’s playback (transmission). The time duration for playback depends on the duty period of the trigger signal and the gate polarity selection (see “ <code>:TRIGger:TYPE:GATE:ACTive</code> ” on page 293). The waveform plays during the inactive state and stops during the active polarity selection state. The active state can be set high or low. The gate mode works only with an external trigger source.

---

**NOTE** The ARB gating behavior described above is opposite to the gating behavior for real-time custom mode.

---

### Example

```
:RAD:DMOD:ARB:TRIG:TYPE GATE
```

The preceding example selects the gate trigger mode.

```
*RST          CONT
```

**Key Entry**            **Continuous**      **Single**      **Gated**

### `:TRIGger:TYPE:CONTInuous[:TYPE]`

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger:TYPE:CONTInuous[:TYPE] FREE|  
TRIGger|RESet  
[:SOURCE]:RADio:DMODulation:ARB:TRIGger:TYPE:CONTInuous[:TYPE]?
```

This commands selects the waveform’s response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see “`:TRIGger:TYPE`” on page 291.

The following list describes the waveform’s response to each of the command choices:

FREE	Turning the ARB format on immediately triggers the waveform. The waveform repeats until you turn the format off, select another trigger, or choose another waveform file.
------	---

- TRIGger**            The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously until you turn the format off, select another trigger, or choose another waveform file.
- RESet**             The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously. Subsequent triggers reset the waveform to the beginning. For a waveform sequence, this means to the beginning of the first segment in the sequence.

**Example**

```
:RAD:DMOD:ARB:TRIG:TYPE:CONT FREE
```

The preceding example selects the continuous trigger free mode.

```
*RST                FREE
```

<b>Key Entry</b>	<b>Free Run</b>	<b>Trigger &amp; Run</b>	<b>Reset &amp; Run</b>
------------------	-----------------	--------------------------	------------------------

**:TRIGger:TYPE:GATE:ACTive**

**Supported**            E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:GATE:ACTive LOW|HIGH
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode.

The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. When the active state occurs, the PSG stops the waveform playback at the last played sample point, then restarts the playback at the next sample point when the inactive state occurs. For more information on triggering and to select gating as the trigger mode, see [“:TRIGger:TYPE” on page 291](#).

The following list describes the PSG’s gating behavior for the polarity selections:

- LOW**                The waveform playback stops when the trigger signal goes low (active state) and restarts when the trigger signal goes high (inactive state).
- HIGH**                The waveform playback stops when the trigger signal goes high (active state) and restarts when the trigger signal goes low (inactive state).

**Example**

```
:RAD:DMOD:ARB:TRIG:TYPE:GATE:ACT HIGH
```

The preceding example sets the active gate state to high.

```
*RST                HIGH
```

<b>Key Entry</b>	<b>Gate Active Low High</b>
------------------	-----------------------------

## :TRIGger[:SOURCE]

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE] KEY|EXT|BUS  
[:SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE]?
```

This command sets the trigger source.

For more information on triggering, see “:TRIGger:TYPE” on page 291. The following list describes the command choices:

**KEY** This choice enables manual triggering by pressing the front panel **Trigger** hardkey.

**EXT** An externally applied signal triggers the waveform. This is the only choice that works with gating. The following conditions affect an external trigger:

- The input connector selected for the trigger signal. You have a choice between the rear panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector. To make the connector selection, see “:TRIGger[:SOURCE]:EXTErnal[:SOURCE]” on page 295.

For more information on the connectors and on connecting the cables, refer to the *User’s Guide*.

- The trigger signal polarity:
  - gating mode, see “:TRIGger:TYPE:GATE:ACTive” on page 293
  - continuous and single modes, see “:TRIGger[:SOURCE]:EXTErnal:SLOPe” on page 296
- The time delay between when the PSG receives a trigger and when the waveform responds to the trigger. There are two parts to setting the delay:
  - setting the amount of delay, see “:TRIGger[SOURCE]:EXTErnal:DELay” on page 295
  - turning the delay on, see “:TRIGger[:SOURCE]:EXTErnal:DELay:STATe” on page 296

**BUS** This choice enables triggering over the GPIB or LAN using the \*TRG or GET commands or the AUXILIARY INTERFACE (RS-232) using the \*TRG command.

### Example

```
:RAD:DMOD:ARB:TRIG EXT
```

The preceding example sets the trigger source to external triggering mode.

```
*RST EXT
```

Key Entry	Trigger Key	Ext	Bus
-----------	-------------	-----	-----

## **:TRIGger[:SOURce]:EXTErnal[:SOURce]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTErnal[:SOURce] EPT1 |
EPT2 | EPTRIGGER1 | EPTRIGGER2
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTErnal[:SOURce]?
```

This command selects which PATTERN TRIG IN connection the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, refer to the **:TRIGger[:SOURce]** command. For more information on the rear panel connectors, refer to the *User's Guide*.

The following list describes the command choices:

- |            |   |
|------------|---|
| EPT1       | This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear panel connector.                         |
| EPT2       | This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector. |
| EPTRIGGER1 | This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear panel connector.                               |
| EPTRIGGER2 | This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear panel AUXILIARY I/O connector.       |

### **Example**

```
:RAD:DMOD:ARB:TRIG:EXT EPT1
```

The preceding example sets the trigger source to the PATTERN TRIG IN rear panel connector.

```
*RST EPT1
```

Key Entry	Patt Trig In 1	Patt Trig In 2
-----------	----------------	----------------

## **:TRIGger[SOURce]:EXTErnal:DELAy**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTErnal:DELAy <val>
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTErnal:DELAy?
```

This command sets the amount of time to delay the PSG's response to an external trigger.

The delay is a path (time) delay between when the PSG receives the trigger and when it responds to the trigger. For example, configuring a trigger delay of two seconds, causes the PSG to wait two seconds after receipt of the trigger before the PSG plays the waveform.

The delay does not occur until it has been enabled On by the **:TRIGger[:SOURce]:EXTErnal:DELAy:STATE** command. The delay value may be set either before or after it has been enabled.

For more information on configuring an external trigger source and to select external as the trigger source, see **"[:TRIGger\[:SOURce\]](#)" on page 294**.

The unit of measurement for the variable <val> is in seconds (nsec–sec).

### Example

```
:RAD:DMOD:ARB:TRIG:EXT:DEL 200MS
```

The preceding example sets the delay for an external trigger to .2 seconds.

```
*RST +1.00000000E-003
```

**Range** 1E-8 to 4E1

**Key Entry** Ext Delay Time

### :TRIGger[:SOURCE]:EXTErnal:DELAy:STATe

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE]:EXTErnal:DELAy:
STATe ON|OFF|1|0
```

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE]:EXTErnal:DELAy:STATe?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, use the `:TRIGger[SOURCE]:EXTErnal:DELAy` command. For more information on configuring an external source, refer to [“:TRIGger\[:SOURCE\]” on page 294](#).

### Example

```
:RAD:DMOD:ARB:TRIG:EXT:DEL 1
```

The preceding example sets the delay active for an external trigger.

```
*RST 0
```

**Key Entry** Ext Delay Off On

### :TRIGger[:SOURCE]:EXTErnal:SLOPe

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE]:EXTErnal:
```

```
SLOPe POSitive|NEGative
```

```
[ :SOURCE]:RADio:DMODulation:ARB:TRIGger[:SOURCE]:EXTErnal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous, single triggering mode. To set the polarity for gating, see [“:TRIGger:TYPE:GATE:ACTIVE” on page 293](#).

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (plays) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, see [“:TRIGger\[:SOURCE\]” on page 294](#).



### Example

```
:RAD:DMOD:ARB:TRIG:EXT:SLOPE POS
```

The preceding example sets the polarity of the active triggering state to positive.

**\*RST**                   NEG

**Key Entry**           **Ext Polarity Neg Pos**

### [ :STATe ]

**Supported**           E8267D with Option 601 or 602

```
[ :SOURce ]:RADio:DMODulation:ARB[ :STATe ] ON|OFF|1|0  
[ :SOURce ]:RADio:DMODulation:ARB[ :STATe ]?
```

This command enables or disables the digital modulation.

ON (1)                This choice sets up the internal hardware to generate the currently selected digital modulation format. When ON is selected, the I/Q state is activated and the I/Q source is set to internal.

OFF (0)               This choice disables the digital modulation capability.

### Example

```
:RAD:DMOD:ARB ON
```

The preceding example turns on the arbitrary waveform generator.

**\*RST**                   0

**Key Entry**           **Digital Modulation Off On**

## Multitone Subsystem—Option 601 or 602 ([:SOURce]:RADio:MTONe:ARB)

### Creating a Multitone Waveform

Use the following steps to create a multitone waveform:

1. Initialize the phase for the multitone waveform (“:SETup:TABLE:PHASe:INITialize” on page 308).
2. Assign the frequency spacing between the tones (“:SETup:TABLE:FSPacing” on page 307).
3. Define the number of tones within the waveform (“:SETup:TABLE:NTONes” on page 308).
4. Modify the power level, phase, and state of any individual tones (“:ROW” on page 309).

### :HEADer:CLEAr

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :MTONe :ARB :HEADer :CLEAr

This command clears the header information from the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the multitone mode must be enabled On by using the [:STATe] command.

**Key Entry** Clear Header

### :HEADer:SAVE

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :MTONe :ARB :HEADer :SAVE

This command saves the header information to the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the multitone must be enabled ON by using the [:STATe] command.

**Key Entry** Save Setup To Header

### :IQ:EXTErnal:FILTer

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :MTONe :ARB :IQ :EXTErnal :FILTer 40e6 | THROugh

[ :SOURce ] :RADio :MTONe :ARB :IQ :EXTErnal :FILTer?

This command selects the filter or through path for I/Q signals routed to the rear panel I and Q outputs. Selecting a filter using this command will automatically set the :IQ:EXTErnal:FILTer:AUTO command to OFF.

40e6 This choice applies a 40 MHz baseband filter.

THROugh This choice bypasses filtering.

### Example

```
:RAD:MTON:ARB:IQ:EXT:FILT 40E6
```

The preceding example selects a 40 MHz filter for the I/Q rear panel signal path.

```
*RST          THR
Key Entry     40.000MHz   Through
```

### **:IQ:EXtErnal:FiLTer:AUTO**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:MTONE:ARB:IQ:EXtErnal:FiLTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:MTONE:ARB:IQ:EXtErnal:FiLTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals routed to the rear panel I/Q outputs. The AUTO feature allows the signal generator to select the filter or through path for the signal.

```
ON(1)          This choice automatically selects the 40 MHz filter optimized for current signal
                generator settings.
OFF(0)         This choice disables the auto feature and allows you to select the 40 MHz filter or
                a through path. Refer to “:IQ:EXtErnal:FiLTer” on page 276 for selecting a filter or
                through path.
```

#### **Example**

```
:RAD:MTON:ARB:IQ:EXT:FILT:AUTO ON
```

The preceding example sets output I/Q filtering to automatic.

```
*RST          1
Key Entry     I/Q Output Filter Manual Auto
```

### **:IQ:MODulation:ATTen**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:MTONE:ARB:IQ:MODulation:ATTen <val>
[:SOURce]:RADio:MTONE:ARB:IQ:MODulation:ATTen?
```

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path. The variable <val> is expressed in decibels (dB).

#### **Example**

```
:RAD:MTON:ARB:IQ:MOD:ATT 20
```

The preceding example sets the modulator attenuator level to 20dB.

```
*RST          +2.00000000E+000
Range         0 to 40 (.01dB resolution)
Key Entry     Modulator Atten Manual Auto
```

## **:IQ:MODulation:ATTen:AUTO**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0  
[:SOURCE]:RADio:MTONe:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the modulator attenuator auto mode. The AUTO mode allows the signal generator to select the best attenuator level for the current settings. The auto mode will be switched to manual if the signal generator receives an AUTO OFF or AUTO 0 command.

- ON (1) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.
- OFF (0) This choice holds the attenuator at its current setting or at the selected value set by the **:IQ:MODulation:ATTen** command.

### **Example**

```
:RAD:MTON:ARB:IQ:MOD:ATT:AUTO OFF
```

The preceding example sets the attenuator in manual mode.

```
*RST 1
```

**Key Entry** **Modulator Atten Manual Auto**

## **:IQ:MODulation:FILTer**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:IQ:MODulation:FILTer 40e6|THROUGH  
[:SOURCE]:RADio:MTONe:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set the **:IQ:MODulation:FILTer:AUTO** command to OFF.

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THROUGH This choice bypasses filtering.

### **Example**

```
:RAD:MTON:ARB:IQ:MOD:FILT THR
```

The preceding example selects a through path for I/Q signals routed to the rear panel outputs.

```
*RST THR
```

**Key Entry** **40.000MHz Through**

## :IQ:MODulation:FILTer:AUTO

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:MTONE:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0  
[:SOURce]:RADio:MTONE:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals modulated onto the RF carrier.

- ON (1) This choice will automatically select the 40 MHz filter optimized for the current signal generator setting.
- OFF (0) This choice disables the automatic filter selection. You may select the 40 MHz filter or the through path by using the [:IQ:MODulation:FILTer](#) command.

### Example

```
:RAD:MTON:ARB:IQ:MOD:FILT:AUTO OFF
```

The preceding example sets the automatic filter selection off.

## :MDEStination:ALCHold

**Supported** E8267D with Option 601 or 602

---

**CAUTION** Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURce]:RADio:MTONE:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4  
[:SOURce]:RADio:MTONE:ARB:MDEStination:ALCHold?
```

This command enables or disables the marker ALC hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, use the [:MPOLarity:MARKer1|2|3|4](#) command. For more information on markers, refer to [“:MARKer:\[SET\]” on page 256](#).

---

**NOTE** Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings.

---

For more information on the marker ALC hold function, refer to the *User’s Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLear” on page 254.
- For clearing all marker points, see “:MARKer:CLear:ALL” on page 255.
- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For setting marker points, see “:MARKer:[SET]” on page 256.

NONE This terminates the marker ALC hold function.

M1–M4 These are the marker choices. The ALC hold feature uses only one marker at a time.

### Example

```
:RAD:MTON:ARB:MDES:ALCH M1
```

The preceding example routes marker one to the ALC hold function.

```
*RST NONE
```

Key Entry	None	Marker 1	Marker 2	Marker 3	Marker 4
-----------	------	----------	----------	----------	----------

### :MDEStination:PULSe

**Supported** E8267D with Option 601 or 602

---

**CAUTION** The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURCE]:RADio:MTONE:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4  
[:SOURCE]:RADio:MTONE:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

---

**NOTE** Do not use ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

---

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker’s polarity, see

“:MPOLarity:MARKer1|2|3|4” on page 303. For more information on setting markers, see “:MARKer:[SET]” on page 256.

---

**NOTE** Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

---

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLEar” on page 254.
- For clearing all marker points, see “:MARKer:CLEar:ALL” on page 255.
- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For setting marker points, see “:MARKer:[SET]” on page 256.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

---

For more information on the marker RF blanking function, refer to the *User’s Guide*.

NONE This terminates the marker RF blanking/pulse function.

M1–M4 These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

**Example**

```
:RAD:MTON:ARB:MDES:PULSE M1
```

The preceding example routes marker one to the Pulse/RF Blanking function.

```
*RST NONE
```

<b>Key Entry</b>	<b>None</b>	<b>Marker 1</b>	<b>Marker 2</b>	<b>Marker 3</b>	<b>Marker 4</b>
------------------	-------------	-----------------	-----------------	-----------------	-----------------

**:MPOLarity:MARKer1|2|3|4**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive  

[:SOURCE]:RADio:MTONe:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLEar” on page 254.
- For clearing all marker points, see “:MARKer:CLEar:ALL” on page 255.
- For shifting marker points, see “:MARKer:ROtate” on page 255.
- For information on markers and setting marker points, see “:MARKer:[SET]” on page 256.

### Example

```
:RAD:MTON:ARB:MPOL:MARK1 NEG
```

The preceding example sets the polarity for marker one to negative.

<b>*RST</b>	POS			
<b>Key Entry</b>	<b>Marker 1 Polarity Neg Pos</b>	<b>Marker 2 Polarity Neg Pos</b>	<b>Marker 3 Polarity Neg Pos</b>	<b>Marker 4 Polarity Neg Pos</b>

### :REFerence:EXTernal:FREQuency

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:REFerence:EXTernal:FREQuency <val>  
[:SOURCE]:RADio:MTONe:ARB:REFerence:EXTernal:FREQuency?
```

This command allows you to enter the frequency of an external reference. The variable <val> is expressed in hertz (Hz–MHz). The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear panel connector. To specify EXTernal as the ARB reference source type, use the :REFerence[:SOURCE] command.

### Example

```
:RAD:MTON:ARB:REF:EXT:FREQ 500KHZ
```

The preceding example sets the external reference to .5 megahertz.

<b>*RST</b>	+1.00000000E+007
<b>Range</b>	2.5E5 to 1E8
<b>Key Entry</b>	<b>Reference Freq</b>

### :REFerence[:SOURCE]

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:REFerence[:SOURCE] INTernal|EXTernal  
[:SOURCE]:RADio:MTONe:ARB:REFerence[:SOURCE]?
```

This command selects either an internal or external reference for the waveform clock. If EXTernal is selected, the external frequency value *must* be entered and the clock signal must be applied to the BASEBAND GEN REF IN rear panel connector. The external reference frequency is entered using the :REFerence:EXTernal:FREQuency command.



**Example**

```
:RAD:MTON:ARB:REF EXT
```

The preceding example sets an external reference as the waveform clock.

```
*RST INT
```

**Key Entry**           **ARB Reference Ext Int**

**:SCLock:RATE**

**Supported**           E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB:SCLock:RATE <sample_clock_rate>
[:SOURCE]:RADio:MTONE:ARB:SCLock:RATE?
```

This command sets the ARB sample clock rate.

The multitone generator should be On before executing this command. If this command is executed before the multitone generator is activated by the [:STATE] command, the entered value will be overridden by a calculated factory default value.

**Example**

```
:RAD:MTON:ARB:SCL:RATE 1E6
```

The preceding example sets the sample clock rate to 1 megahertz.

```
*RST +1.00000000E+006
```

**Range**               1 to 1E8

**Key Entry**           **ARB Sample Clock**

**:SETup**

**Supported**           E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB:SETup "<file_name>"
[:SOURCE]:RADio:MTONE:ARB:SETup?
```

This command retrieves a multitone waveform file from the signal generator's MTONE directory. The directory path is implied in the command and does not need to be specified. After the waveform file is loaded into memory you must send the command to turn on the Multitone generator. For information on the file name syntax, see "File Name Variables" on page 10.

**Example**

```
:RAD:MTON:ARB:SET "Multi_Setup"
```

The preceding example loads the Multi\_Setup waveform file into the signal generator's memory.

**Key Entry**           **Load From Selected File**

## :SETup:STORe

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:SETup:STORe "<file_name>"
```

This command stores the current multitone waveform setup in the signal generator's MTONE directory using the "<file\_name>" file name. The directory path is implied in the command and does not need to be specified.

### Example

```
:RAD:MTON:ARB:SET:STOR "Multi_Setup1"
```

The preceding example stores the current multitone setup to the Multi\_Setup1 file and stores it in the signal generator's MTONE directory.

**Key Entry**           **Store To File**

## :SETup:TABLE

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONe:ARB:SETup:TABLE <freq_spacing>,<num_tones>,  
{<phase>,<state>}  
[:SOURCE]:RADio:MTONe:ARB:SETup:TABLE?
```

This command creates and configures a multitone waveform. The frequency offset, power, phase, and state value are returned when a query is initiated. The parameter format is as follows:

<freq\_spacing>       Spacing is limited by the 80 MHz bandwidth of the arbitrary waveform generator and the number of tones desired. No units are specified.

<num\_tones>           There must be a minimum of two tones and a maximum of 64.

<phase>               0 to 359

<state>               An enabled state is +1. A disabled state is 0.

---

**NOTE** Frequency offset is related to frequency spacing. Frequency offset between tones equals the frequency spacing.

---

To set the frequency spacing, refer to the [:SETup:TABLE:FSPacing](#) command. To set the power level for tones refer to the [:ROW](#) command.

### Example

```
:RAD:MTON:ARB:SET:TABL 1000000,3,90,1,60,0,45,1
```

The preceding example creates a multitone setup consisting of 3 tones with 1 megahertz tone spacing. The first tone phase is 90 degrees and the state is on. The second tone phase is 60 degrees and the state is off. The third tone phase is 45 degrees and the state is on.

*RST	Tone	<frequency offset>	<power>	<phase>	<state>
	Tone 1	-35000	+0.00000000E+000	+0	+1
	Tone 2	-25000	+0.00000000E+000	+0	+1
	Tone 3	-15000	+0.00000000E+000	+0	+1
	Tone 4	-5000	+0.00000000E+000	+0	+1

*RST	Tone	<frequency offset>	<power>	<phase>	<state>
	Tone 5	+5000	+0.00000000E+000	+0	+1
	Tone 6	+15000	+0.00000000E+000	+0	+1
	Tone 7	+25000	+0.00000000E+000	+0	+1
	Tone 8	+35000	+0.00000000E+000	+0	+1

**Range**                    <freq\_spacing> (2 tones): 1E4 to 8E7      <num\_tones>: 2 to 64  
                              <freq\_spacing> (>2 tones): 1E4 to (80 MHz ÷ (num\_tones - 1))  
                              <phase>: 0 to 359

**Key Entry**                **Freq Spacing**            **Number Of Tones**            **Toggle State**

### :SETup:TABLE:FSPacing

**Supported**                E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB:SETup:TABLE:FSPacing <freq_spacing>
[ :SOURCE]:RADio:MTONE:ARB:SETup:TABLE:FSPacing?
```

This command sets the frequency spacing between tones. The variable <freq\_spacing> is expressed in hertz (Hz–MHz) and is limited to the 80 megahertz bandwidth of the arbitrary waveform generator.

To set frequency spacing and additional parameters required to create or configure a multitone waveform, refer to [:SETup:TABLE](#) command. This command is the second step in creating a multitone waveform. The four steps required when [Creating a Multitone Waveform](#) are listed in this command.

#### Example

```
:RAD:MTON:ARB:SET:TABL:FSP 100KHZ
```

The preceding example sets a 100 kHz frequency spacing between tones.

\*RST                        +1.00000000E+004

**Range**                    <freq\_spacing> (2 tones): 100 Hz to 80 MHz  
                              <freq\_spacing> (>2 tones): 1E2 to (80 MHz ÷ (num\_tones - 1))

**Key Entry**                **Freq Spacing**

## **:SETup:TABLE:NTONes**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB:SETup:TABLE:NTONes <num_tones>  
[:SOURCE]:RADio:MTONE:ARB:SETup:TABLE:NTONes?
```

This command defines the number of tones in the multitone waveform. To specify the number of tones and additional parameters required to create or configure a multitone waveform, use the [:SETup:TABLE](#) command. This is the third step in creating a multitone waveform. The four steps required when [Creating a Multitone Waveform](#) are listed in this command.

### **Example**

```
:RAD:MTON:ARB:SET:TABL:NTON 4
```

The preceding example sets four tones in the current multitone table.

**\*RST** +8

**Range** 2 to 64

**Key Entry** **Number Of Tones**

## **:SETup:TABLE:PHASe:INITialize**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB:SETup:TABLE:PHASe:INITialize FIXed|RANdom  
[:SOURCE]:RADio:MTONE:ARB:SETup:TABLE:PHASe:INITialize?
```

This command initializes the phase in the multitone waveform table.

**FIXed** This choice sets the phase of all tones to the fixed value of 0 degrees.

**RANdom** This choice sets the phase of all tones to random values based on the setting on the random seed generator.

To change the random number generator seed value, refer to the [:SETup:TABLE:PHASe:INITialize:SEED](#) command.

This command is the first of four steps in creating a multitone waveform ([page 298](#)).

### **Example**

```
:RAD:MTON:ARB:SET:TABL:PHAS:INIT RAND
```

The preceding example sets the phase for the tones to a random number.

**\*RST** FIX

**Key Entry** **Initialize Phase Fixed Random**

## **:SETup:TABLE:PHASe:INITialize:SEED**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE ] :RADio:MTONe:ARB:SETup:TABLE:PHASe:INITialize:SEED FIXed | RANDom
[ :SOURCE ] :RADio:MTONe:ARB:SETup:TABLE:PHASe:INITialize:SEED?
```

This command initializes the random number generator seed that is used to generate phase values for the multitone waveform tones.

**FIXed** This choice sets the random number generator seed to a fixed value. This selection will generator random and repeatable phase values: the same phase values will be generated with subsequent execution of the command.

**RANDom** This choice sets the random number generator seed to a random value. This changes the phase value after each initialization of the phase.

### **Example**

```
:RAD:MTON:ARB:SET:TABL:PHAS:INIT:SEED RAND
```

The preceding example sets the random number generator seed to a random value.

```
*RST FIX
```

**Key Entry** **Random Seed Fixed Random**

## **:ROW**

**Supported** E8267D with Option 601 or 602

```
[ :SOURCE ] :RADio:MTONe:ARB:SETup:TABLE:ROW <row_number> , <power> ,
<phase> , <state>
[ :SOURCE ] :RADio:MTONe:ARB:SETup:TABLE:ROW? <row_number>
```

This command modifies the indicated tone (row) of the multitone waveform.

**<row\_number>** The number of rows for this variable is determined by the [:SETup:TABLE](#) command.

**<power>** The power level of the tone defined in the row number. Power levels for all tones must not exceed the power level of the signal generator. The power variable is expressed in decibels (dB)

**<phase>** The phase of the tone relative to the carrier. The phase variable is expressed in degrees.

**<state>** The state of the tone in this row can be enabled or disabled.

Frequency offset, power, phase, and state value are returned when a query is initiated. The output format is as follows:

```
<frequency_offset> , <power> , <phase> , <state>
```

The [:SETup:TABLE](#) command provides information on how to change the number of rows. This command is the final step in creating a multitone waveform ([page 298](#)).

### Example

```
:RAD:MTON:ARB:SET:TABL:ROW 2,-10,40,0
```

The preceding example modifies row number two in the currently selected multitone table. The power is set to -10 dB, the phase is set to 40 degrees, and the state is off.

```
*RST          frequency offset: -3.50000000E+004      <power>: +0.00000000E+000  
              <phase>: +0.00000000E+000      <state>: 1
```

```
Range          frequency offset: -4E7 to 4E7      <power>: -80 to 0      <phase>: 0 to 359  
              <state>: 1
```

**Key Entry**            **Goto Row**        **Toggle State**

### [ :STATe ]

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:MTONE:ARB[ :STATe] ON|OFF|1|0  
[:SOURCE]:RADio:MTONE:ARB[ :STATe]?
```

This command enables or disables the operating state of the multitone waveform generator.

### Example

```
:RAD:MTON:ARB ON
```

The preceding example turns on the multitone generator.

```
*RST          0
```

**Key Entry**            **Multitone Off On**

## Real Time GPS Subsystem—Option 409 ([:SOURCE]:RADIo[1]|2|3|4:GPS)

### :DATA

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:DATA PN9|PN15|FIX4| "<user file>"
[:SOURCE]:RADIo:GPS:DATA?
```

This command sets the data type for the selected data mode.

**\*RST** PN9

**Key Entry** PN9 PN15 FIX4 User file

**Remarks** This command is effective only when the data mode is RAW or ENCOded. To set the data mode, refer to :DMODE.

### :DMODE

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:DMODE RAW|ENCOded|TLM
[:SOURCE]:RADIo:GPS:DMODE?
```

This command sets the data mode.

**RAW** This choice modulates data onto the C/A (coarse acquisition) code at 50-bits per second. No parity bits are computed by the signal generator. Every 6 seconds, 300-bits from the source data are transmitted.

**ENCOded** This choice modulates data onto the C/A (coarse acquisition) code at 50-bits per second. The signal generator computes 6 parity bits for every 24 data bits from the selected data source. Every six seconds, 240-bits of the source data are transmitted along with 60 computed parity bits.

**TLM** This choice transmits a standard default navigation data transmission which includes a telemetry word (TLM), a handover word (HOW), and default navigation data. The signal generator transmits an incrementing time-of-week (TOW) as part of the HOW.

**\*RST** RAW  
**Key Entry** Data Mode Raw Enc TLM  
**Remarks** Since the TLM mode transmits default navigation data, there is no data selection for this mode.  
For selecting the data type when RAW or ENCOded is the selection, refer to “:DATA” on page 311.

## :DSHift

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:DSHift <val>  
[:SOURCE]:RADIo:GPS:DSHift?
```

This command sets the frequency and chip rate offsets to simulate a doppler shift. The variable <val> is expressed in units of hertz (Hz to kHz).

**\*RST** +0.00000000E+000  
**Range** -125kHz to 125kHz  
**Key Entry** Doppler Shift

**Remarks** The lower bound of the doppler shift is limited by the frequency set on the signal generator. For example, if the signal generator frequency is set to 100 kHz, then the lower limit of the doppler shift would be 0.00 Hz. The doppler shift can not extend lower than the limitations of the signal generator

## :FILTer

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:FILTer RNYQuist|NYQuist|GAUSSian|RECTangle|  
IS95|IS95_EQ|IS95_MOD|IS95_MOD_EQ|AC4Fm|UGGaussian| "<user FIR>"  
[:SOURCE]:RADIo:GPS:FILTer?
```

This command sets the pre-modulation filter type.

**IS95** This choice selects a filter that meets the criteria of the IS-95 standard.  
**IS95\_EQ** This choice selects a filter which is a combination of the IS-95 filter (above) and the equalizer filter described in the IS-95 standard. This filter is only used for IS-95 baseband filtering.



IS95\_MOD This choice selects a filter that meets the criteria of the IS-95 error function (for improved adjacent channel performance) with lower passband rejection than the filter specified in the IS-95 standard.

IS95\_MOD\_EQ This choice selects a filter which is a combination of the equalizer filter described in the IS-95 standard and a filter that meets the criteria of the IS-95 error function (for improved adjacent channel performance) with lower passband rejection.

AC4Fm This choice selects a predefined Association of Public Safety Communications Officials (APCO) specified compatible 4-level frequency modulation (C4FM) filter.

UGGaussian This choice selects a backwards compatible GSM Gaussian filter (Gaussian filter with a fixed BbT value of 0.300) for the ESG E44xxB Option UN3 or UN4.

"<user FIR>" This variable is any filter file that you have stored into memory.

**\*RST**                   RECT

<b>Key Entry</b>	<b>Root Nyquist</b>	<b>Nyquist</b>	<b>Gaussian</b>	<b>Rectangle</b>	<b>IS-95</b>	<b>IS-95 w/EQ</b>
	<b>IS-95 Mod</b>	<b>IS-95 Mod w/EQ</b>	<b>APC025 C4FM</b>		<b>UN3/4 GSM Gaussian</b>	
	<b>User FIR</b>					

## **:FILTer:ALPHa**

**Supported**           E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:FILTer:ALPHa <val>
[:SOURCE]:RADIo:GPS:FILTer:ALPHa?
```

This command sets the Nyquist or root Nyquist filter's alpha value.

The filter alpha value can be set to the minimum value (0), maximum value (1), or in between by using fractional numeric values (0.001–0.999).

**\*RST**                   +2.20000000E-001

**Range**                0.000–1.000

**Key Entry**           **Filter Alpha**

**Remarks**           To change the current filter type, refer to “:FILTer” on page 312.

## :FILTer:BBT

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:GPS:FILTer:BBT <val>  
[ :SOURCE]:RADio:GPS:FILTer:BBT?
```

This command changes the bandwidth-multiplied-by-bit-time (BbT) filter parameters.

The filter BbT value can be set to the minimum value (0), the maximum value (1), or in between by using fractional numeric values (0.001–0.999)

**\*RST** +5.00000000E-001

**Range** 0.000–1.000

**Key Entry** **Filter BbT**

**Remarks** This command is effective only after choosing a Gaussian filter. It does not have an effect on other types of filters.

To change the current filter type, refer to “:FILTer” on page 312.

## :FILTer:CHANnel

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:GPS:FILTer:CHANnel EVM|ACP [ :SOURCE]:RADio:GPS:FILTer:CHANnel?
```

This command optimizes the Nyquist and root Nyquist filters to minimize error vector magnitude (EVM) or to minimize adjacent channel power (ACP).

**EVM** This choice provides the most ideal passband.

**ACP** This choice improves stopband rejection.

**\*RST** EVM

**Key Entry** **Optimize FIR For EVM ACP**

**Remarks** To change the current filter type, refer to “:FILTer” on page 312.

## :IQPhase

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:IQPhase NORMAL|INVerted
[:SOURCE]:RADIo:GPS:IQPhase?
```

This command sets the I/Q phase for the GPS signal.

**NORMAL** This choice selects normal phase polarity.

**INVerted** This choice inverts the internal Q signal.

**\*RST** NORM

**Key Entry** IQ Phase Normal Invert

## :PCODE

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:PCODE <val>
[:SOURCE]:RADIo:GPS:PCODE?
```

This command sets the P code power relative to the C/A code power.

**\*RST** -3.00000000E+000

**Range** -40 to 0

**Key Entry** P Code Pwr

**Remarks** This command is normally used when the CAP (C/A+P) ranging mode choice is selected. Refer to [:RCODE](#) for selecting the ranging mode.

## :RCODE

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADIo:GPS:RCODE CA|P|CAP
[:SOURCE]:RADIo:GPS:RCODE?
```

This command selects the ranging code for the GPS transmission.

**CA** This choice selects a 1023-bit pseudorandom C/A (coarse acquisition) code that is BPSK modulated onto the L1 (1575.42 MHz) carrier. The C/A code factory set chip rate is 1.023 Mcps using a 10.23 Mcps reference clock.

<b>P</b>	This choice selects the precise (P) code which is a very long pseudorandom sequence that is BPSK modulated onto the L2 (1227.6 MHz) carrier. The P code factory set chip rate is 10.23 Mcps using a 10.23 Mcps reference clock.
<b>CAP</b>	This choice permits both the C/A (coarse acquisition) and P (precise) codes to modulate the L1 (1575.42 MHz) carrier simultaneously by providing the P code on the Q component and the C/A code in quadrature on the I component.
<b>*RST</b>	CA
<b>Key Entry</b>	<b>Ranging Code C/A P C/A+P</b>

## **:REFClk**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:GPS:REFClk INT|Ext  
[:SOURCE]:RADio:GPS:REFClk?
```

This command sets the GPS reference clock to either internal or external.

<b>INT</b>	This selection sets the signal generator to use the internal chip clock.
<b>EXT</b>	This selection sets the signal generator to use an external chip clock which is supplied to the DATA CLOCK INPUT connector.
<b>*RST</b>	INT
<b>Key Entry</b>	<b>GPS Ref Clk</b>

## **:REFFreq**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:GPS:REFFreq <val><unit>  
[:SOURCE]:RADio:GPS:REFFreq?
```

This command sets the GPS reference clock frequency. If an external source is being used, its frequency must match the value set with this command

<b>*RST</b>	+1.02300000E+007
<b>Range</b>	1kCPS–12.5MCPS
<b>Key Entry</b>	<b>GPS Ref (f0)</b>
<b>Remarks</b>	Changing the GPS reference frequency will change the P and C/A code chip rates.

## **:SATid**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:GPS:SATid <val>
```

[:SOURCE]:RADio:GPS:SATid?

This command selects the pseudorandom number (PRN) code used for transmission.

Satellite identification numbers 1–32 are used for GPS satellites. Satellite identification numbers 33–37 are reserved for ground transmitter use in the real-world system.

**\*RST**                    +1  
**Range**                    1–37  
**Key Entry**                **Satellite ID**

### **[ :STATE ]**

**Supported**                E8267D with Option 409

[:SOURCE]:RADio:GPS[:STATE] ON|OFF|1|0  
[:SOURCE]:RADio:GPS[:STATE]?

This command enables or disables the real-time GPS signal.

**\*RST**                    0  
**Key Entry**                **Real-time GPS Off On**

## Real Time MSGPS Subsystem—Option 409 ([:SOURce]:RADio[1]|2|3|4:MSGPs)

### :IQPHase

**Supported** E8267D with Option 409

```
[ :SOURce]:RADio:MSGPs:IQPHase NORMal|INVerted  
[:SOURce]:RADio:MSGPs:IQPHase?
```

This command sets the I/Q phase for the MSGPS signal.

**NORMal** This choice selects normal phase polarity.

**INVerted** This choice inverts the internal Q signal.

**\*RST** NORM

**Key Entry** IQ Phase Normal Invert

### :PLAYmode

**Supported** E8267D with Option 409

```
[ :SOURce]:RADio:MSGPs:PLAYmode RUN|PAUSE  
[:SOURce]:RADio:MSGPs:PLAYmode?
```

This command pauses or plays the real-time MSGPS scenario.

**\*RST** RUN

**Key Entry** Pause/Resume

## :REFClk

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:REFClk INTERNAL|EXTERNAL  
[:SOURCE]:RADio:MSGPs:REFClk?
```

This command sets the MSGPS reference clock to either internal or external.

**INTERNAL** This selection sets the signal generator to use the internal chip clock.

**EXTERNAL** This selection sets the signal generator to use an external chip clock which is supplied to the DATA CLOCK INPUT connector.

**\*RST** INT

**Key Entry** **GPS Ref Clk**

## :REFFreq

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:REFFreq <val><unit>  
[:SOURCE]:RADio:MSGPs:REFFreq?
```

This command sets the MSGPS reference clock frequency. If an external reference clock is being used, its frequency must match the value set with this command

**\*RST** +1.02300000E+007

**Range** 1.023Mcps  $\pm$ 10%

**Key Entry** **GPS Ref (f0)**

**Remarks** Changing the GPS reference frequency will change the C/A code chip rate.

## :REStart

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:REStart
```

This command sets the real-time MSGPS scenario to the beginning.

**Key Entry** **Restart**

## **:SCENario**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:SCENario "<file_name>"  
[:SOURCE]:RADio:MSGPs:SCENario?
```

This command selects the real-time MSGPS scenario to play.

**Key Entry**           **Select Scenario**

## **:SCENario:SATellites**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:SCENario:SATellites <val>  
[:SOURCE]:RADio:MSGPs:SCENario:SATellites?
```

This command sets the number of satellites in view to include in the generated MSGPS signal.

**Key Entry**           **Number of Satellites**

## **:SCENario:STATUS**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs:SCENario:STATUS?
```

This query returns the following information for the currently selected scenario as a comma-separated list:

Scenario date, scenario time, scenario position, scenario length, satellite IDs

**Key Entry**           **Scenario**

## **[:STATe]**

**Supported** E8267D with Option 409

```
[ :SOURCE]:RADio:MSGPs[:STATe] ON|OFF|1|0  
[:SOURCE]:RADio:MSGPs[:STATe]?
```

This command enables or disables the real-time MSGPS signal.

**\*RST**                0

**Key Entry**           **Real-time MSGPS Off On**



## Two Tone Subsystem ([:SOURce]:RADio:TTONe:ARB)

### :ALIGNment

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:TTONe:ARB:ALIGNment LEFT|CENTer|RIGHT  
[:SOURce]:RADio:TTONe:ARB:ALIGNment?
```

This command will align the two tones either left, center or right of the carrier frequency.

#### Example

```
:RAD:TTON:ARB:ALIG CENT
```

The preceding example aligns each of the two tones equidistant from the carrier frequency.

**Key Entry** Alignment Left Cent Right

### :APPLY

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:TTONe:ARB:APPLY
```

This command will cause the two-tone waveform to be regenerated using the current settings.

This command has no effect unless the two-tone waveform generator is enabled and a change has been made to the frequency spacing setting.

**Key Entry** Apply Settings

### :FSPacing

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:TTONe:ARB:FSPacing <freq_spacing>  
[:SOURce]:RADio:TTONe:ARB:FSPacing?
```

This command sets the frequency spacing between the tones.

The variable <freq\_spacing> is expressed in hertz (Hz–MHz).

#### Example

```
:RAD:TTON:ARB:FSP 10MHZ
```

The preceding example sets a 10 megahertz frequency spacing for the two tones.

**\*RST** +1.00000000E+004

**Range** 1E2 to 8E7

**Key Entry** Freq Separation

## **:HEADer:CLEAr**

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :TTONe :ARB :HEADer :CLEAr

This command clears the header information from the header file used for the two-tone waveform format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the two tone generator must be enabled ON by the [:STATe] command.

**Key Entry** Clear Header

## **:HEADer:SAVE**

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :TTONe :ARB :HEADer :SAVE

This command saves the header information to the header file used for the two-tone waveform format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *User's Guide* for information on header files.

For this command to function, the two tone generator must be enabled ON by the [:STATe] command.

**Key Entry** Save Setup To Header

## **:IQ:EXTErnal:FILTer**

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :TTONe :ARB :IQ :EXTErnal :FILTer 40e6 | THROugh

[ :SOURce ] :RADio :TTONe :ARB :IQ :EXTErnal :FILTer?

This command selects the filter or through path for I/Q signals routed to the rear panel I and Q outputs. Selecting a filter with this command automatically sets the :IQ:EXTErnal:FILTer:AUTO command to OFF.

40e6 This choice applies a 40 MHz baseband filter.

THROugh This choice bypasses filtering.

### **Example**

```
:RAD:TTON:ARB:IQ:EXT:FILT THR
```

The preceding example sets the through path for I/Q signal.

\*RST THR

**Key Entry** 40.000MHz Through

## **:IQ:EXTErnal:FILTer:AUTO**

**Supported** E8267D with Option 601 or 602

[ :SOURce ] :RADio :TTONe :ARB :IQ :EXTErnal :FILTer :AUTO ON | OFF | 1 | 0

```
[ :SOURCE]:RADIO:TTONE:ARB:IQ:EXTERNAL:FILTER:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals routed to the rear panel I/Q outputs.

- ON(1) This choice automatically selects the 40 MHz filter optimized for the current signal generator settings.
- OFF(0) This choice disables the auto feature and allows you to select the 40 MHz filter or a through path using the `:IQ:EXTERNAL:FILTER` command.

### Example

```
:RAD:TTON:ARB:IQ:EXT:FILT:AUTO ON
```

The preceding example enables the automatic filter selection.

```
*RST 1
```

**Key Entry**            **I/Q Output Filter Manual Auto**

### :IQ:MODulation:ATTen

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:TTONE:ARB:IQ:MODulation:ATTen <val><unit>
[:SOURCE]:RADIO:TTONE:ARB:IQ:MODulation:ATTen?
```

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path. The variable <val> is expressed in decibels (dB).

### Example

```
:RAD:TTON:ARB:IQ:MOD:ATT 20
```

The preceding example sets the modulator attenuator to 20 dB.

```
*RST +2.00000000E+000
```

**Range**                0 to 40 dB

**Key Entry**            **Modulator Atten Manual Auto**

### :IQ:MODulation:ATTen:AUTO

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADIO:TTONE:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURCE]:RADIO:TTONE:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the modulator attenuator auto mode. The auto mode will be switched to manual if the signal generator receives an AUTO OFF or AUTO ON command.

- ON (1) This choice enables the attenuation auto mode which allows the signal generator to select the attenuation level that optimizes performance based on the current conditions.
- OFF (0) This choice holds the attenuator at its current setting or at the selected value set by the `:IQ:MODulation:ATTen` command.

### Example

```
:RAD:TTON:ARB:IQ:MOD:ATT:AUTO ON
```

The preceding example enables the attenuator automatic mode.

```
*RST 1
```

**Key Entry**            **Modulator Atten Manual Auto**

### :IQ:MODulation:FILTer

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:TTONe:ARB:IQ:MODulation:FILTer 40e6|THrough  
[:SOURCE]:RADio:TTONe:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set the [:IQ:MODulation:FILTer:AUTO](#) command to Off.

40E6                    This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough                This choice bypasses filtering.

### Example

```
:RAD:TTON:ARB:IQ:MOD:FILT 40E6
```

The preceding example selects the 40 MHz filter.

```
*RST THR
```

**Key Entry**            **40.000MHz**        **Through**

### :IQ:MODulation:FILTer:AUTO

**Supported**            E8267D with Option 601 or 602

```
[ :SOURCE]:RADio:TTONe:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0  
[:SOURCE]:RADio:TTONe:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic filter selection for I/Q signals modulated onto the RF carrier.

ON (1)                    This choice will automatically select the 40 MHz filter optimized for the current signal generator setting.

OFF (0)                    This choice disables the automatic filter selection. The digital modulation filter or through path is selected by the [:IQ:MODulation:FILTer](#) command.

### Example

```
:RAD:TTON:ARB:IQ:MOD:FILT:AUTO ON
```

The preceding example enables the automatic filter selection for I/Q signals.

```
*RST 1
```

**Key Entry**            **I/Q Mod Filter Manual Auto**

### :MDEStination:ALCHold

**Supported**            E8267D with Option 601 or 602

---

**CAUTION**    Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURce]:RADio:TTONE:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4  
[:SOURce]:RADio:TTONE:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see [:MPOLarity:MARKer1|2|3|4](#) command. For more information on markers, see [“:MARKer:\[SET\]” on page 256](#).

---

**NOTE**    Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

---

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE**    A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

---

For more information on the marker ALC hold function, refer to the *User's Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see [“:MARKer:CLEar” on page 254](#).
- For clearing all marker points, see [“:MARKer:CLEar:ALL” on page 255](#).

- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For setting marker points, see “:MARKer:[SET]” on page 256.

NONE This terminates the marker ALC hold function.

M1–M4 These are the marker choices. The ALC hold feature uses only one marker at a time.

### Example

```
:RAD:TTON:ARB:MDES:ALCH M2
```

The preceding example routes marker two to the ALC hold function.

```
*RST NONE
```

**Key Entry**            **None**   **Marker 1**   **Marker 2**   **Marker 3**   **Marker 4**

### :MDEStination:PULSe

**Supported**            E8267D with Option 601 or 602

---

**CAUTION**    The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unlevelled condition that may create a spike in the RF output, potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

---

```
[ :SOURce]:RADio:TTONe:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4  
[:SOURce]:RADio:TTONe:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

---

**NOTE**    Do not use ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

---

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker’s polarity, see :MPOlarity:MARKer1|2|3|4 command. For more information on markers, see “:MARKer:[SET]” on page 256.

---

**NOTE**    Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

---

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLEar” on page 254.
- For clearing all marker points, see “:MARKer:CLEar:ALL” on page 255.
- For shifting marker points, see “:MARKer:ROTate” on page 255.

- For setting marker points, see “:MARKer:[SET]” on page 256.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

---

**NOTE** A waveform file that has unspecified settings in the file header uses the previous waveform’s routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

---

For more information on the marker RF blanking function, refer to the *User’s Guide*.

NONE This terminates the marker RF blanking/pulse function.  
M1–M4 These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

**Example**

```
:RAD:TTON:ARB:MDES:ALCH M3
```

The preceding example routes marker three to the Pulse/RF Blanking function.

```
*RST NONE
```

<b>Key Entry</b>	<b>None</b>	<b>Marker 1</b>	<b>Marker 2</b>	<b>Marker 3</b>	<b>Marker 4</b>
------------------	-------------	-----------------	-----------------	-----------------	-----------------

**:MPOLarity:MARKer1|2|3|4**

**Supported** ES267D with Option 601 or 602

```
[ :SOURCE]:RADio:TTONE:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive  
[:SOURCE]:RADio:TTONE:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see “:MARKer:CLEar” on page 254.
- For clearing all marker points, see “:MARKer:CLEar:ALL” on page 255.
- For shifting marker points, see “:MARKer:ROTate” on page 255.
- For information on markers and setting marker points, see “:MARKer:[SET]” on page 256.

**Example**

```
:RAD:TTON:ARB:MPOL:MARK1 POS
```

The preceding example sets the polarity for marker one to positive.

```
*RST POS
```

<b>Key Entry</b>	<b>Marker 1 Polarity Neg Pos</b>	<b>Marker 2 Polarity Neg Pos</b>	<b>Marker 3 Polarity Neg Pos</b>
------------------	----------------------------------	----------------------------------	----------------------------------

## Marker 4 Polarity Neg Pos

### :REfERENCE:EXtERnal:FREQUency

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ] :RADio :TTONe :ARB :REfERENCE :EXtERnal :FREQUency <val>  
[ :SOURce ] :RADio :TTONe :ARB :REfERENCE :EXtERnal :FREQUency ?
```

This command allows you to enter the frequency of the external reference.

The variable <val> is expressed in hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear panel connector.

To specify external as the ARB reference source type, refer to “:REfERENCE[:SOURce]” on page 263.

#### Example

```
:RAD:TTON:ARB:REF:EXT:FREQ 1MHZ
```

The preceding example sets the external reference to 1 megahertz.

**\*RST** +1.00000000E+007

**Range** 2.5E5 to 1E8

**Key Entry** Reference Freq

### :REfERENCE[:SOURce]

**Supported** E8267D with Option 601 or 602

```
[ :SOURce ] :RADio :TTONe :ARB :REfERENCE [ :SOURce ] INTernal | EXtERnal  
[ :SOURce ] :RADio :TTONe :ARB :REfERENCE [ :SOURce ] ?
```

This command selects either an internal or external reference for the waveform clock. If EXtERnal is selected, the external frequency value *must* be entered and the clock signal must be applied to the BASEBAND GEN REF IN rear panel connector. The external reference frequency is entered using the :REfERENCE:EXtERnal:FREQUency command.

#### Example

```
:RAD:TTON:ARB:REF EXT
```

The preceding example sets an external reference as the waveform clock.

**\*RST** INT

**Key Entry** ARB Reference Ext Int



## **:SCLock:RATE**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:TTONE:ARB:SCLock:RATE <sample_clock_rate>
[:SOURce]:RADio:TTONE:ARB:SCLock:RATE?
```

This command sets the ARB sample clock rate.

The multitone generator should be on before executing this command. If this command is executed before the multitone generator is activated by the [:STATe] command, the entered value will be overridden by a calculated factory default value.

### **Example**

```
:RAD:TTON:ARB:SCL:RATE 1MHZ
```

The preceding example sets the ARB sample clock to 1 MHz.

```
*RST +1.00000000E+008
```

**Range** 1 to 1E8

**Key Entry** **ARB Sample Clock**

## **[:STATe]**

**Supported** E8267D with Option 601 or 602

```
[ :SOURce]:RADio:TTONE:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:TTONE:ARB[:STATe]?
```

This command enables or disables the on/off operational state of the two-tone waveform generator function.

### **Example**

```
:RAD:TTON:ARB ON
```

The preceding example turns on the two-tone generator.

```
*RST 0
```

**Key Entry** **Two Tone Off On**

## Wideband Digital Modulation Subsystem ([:SOURce]:WDM)

### :IQADjustment:IOFFset

**Supported** E8267D with Option 015

```
[ :SOURce]:WDM:IQADjustment:IOFFset <val><unit>  
[:SOURce]:WDM:IQADjustment:IOFFset?
```

This command sets the I channel offset value, as a percent of the full scale. 100% offset is equivalent to 500 mV DC at the input connector.

#### Example

```
:WDM:IQAD:IOFF 100MV
```

The preceding example sets an offset of 100 mV DC for the I signal.

**\*RST** +0.00000000E+000

**Range** -5E1 to +5E1

**Key Entry** **I Offset**

### :IQADjustment:QOFFset

**Supported** E8267D with Option 015

```
[ :SOURce]:WDM:IQADjustment:QOFFset <val><unit>  
[:SOURce]:WDM:IQADjustment:QOFFset?
```

This command sets the Q channel offset value, as a percent of the full scale. 100% offset is equivalent to 500 mV DC at the input connector

#### Example

```
:WDM:IQAD:QOFF 100MV
```

The preceding example sets an offset of 100 mV DC for the Q signal.

**\*RST** +0.00000000E+000

**Range** -5E1 to +5E1

**Key Entry** **Q Offset**

## **:IQADjustment:QSKew**

**Supported** E8267D with Option 601 or 602 and Option 015

```
[ :SOURce]:WDM:IQADjustment:QSKew <val>
[:SOURce]:WDM:IQADjustment:QSKew?
```

This command adjusts the phase angle between the I and Q vectors.  
The variable <val> is expressed in degrees with a minimum resolution of 0.1.

Positive skew increases the angle from 90 degrees while negative skew decreases the angle from 90 degrees. When the quadrature skew is zero, the phase angle is 90 degrees. If the signal generator is operating at frequencies greater than 3.3 GHz, quadrature skew settings greater than  $\pm 5$  degrees will not be within specifications.

This command is effective only if the state of the I/Q adjustment function is set to ON by the [:IQADjustment\[:STATe\]](#) command.

### **Example**

```
:WDM:IQAD:QSK 3.1
```

The preceding example sets the skew value for the Q signal to 3.1 degrees.

**\*RST** +0.00000000E+000

**Range** -1E1 to +1E1

**Key Entry** **Quadrature Skew**

## **:IQADjustment[:STATe]**

**Supported** E8267D with Option 015

```
[ :SOURce]:WDM:IQADjustment[:STATe] ON|OFF|1|0
[:SOURce]:WDM:IQADjustment[:STATe]?
```

This command enables or disables the wideband I/Q adjustments.

### **Example**

```
:WDM:IQAD ON
```

The preceding example enables I/Q adjustments.

**\*RST** 0

**Key Entry** **I/Q Adjustments Off On**

## **:STATe**

**Supported** E8267D with Option 015

```
[ :SOURce]:WDM:STATe ON|OFF|1|0
[:SOURce]:WDM:STATe?
```

This command enables or disables the wideband I/Q modulator.  
The I/Q modulator is automatically enabled whenever a digital modulation form is turned on and when active, the I/Q annunciator appears on the signal generator's display.

**Example**

```
:WDM:STAT ON
```

The preceding example enables the wideband I/Q modulator.

```
*RST 0
```

**Key Entry I/Q Off On**

---

## 6 Digital Signal Interface Module Commands

This chapter provides SCPI descriptions for commands available with Agilent's N5102A Digital Signal Interface Module. Refer to the *User's Guide, E8257D/67D, E8663D PSG Key Reference*, or *N5101A Digital Signal Interface Module Installation Guide* for more information on the N5102A interface module.

- “Digital Subsystem ([:SOURce])” on page 333

### Digital Subsystem ([:SOURce])

#### :DIGital:CLOCK:CPS

**Supported** E8267D Option 601 or 602 with Option 003

```
[:SOURce]:DIGital:CLOCK:CPS 1|2|4
```

```
[:SOURce]:DIGital:CLOCK:CPS?
```

This command selects the number of clock cycles per sample. The command is used with parallel or parallel interleaved port configurations. If this command is executed with a serial port configuration or an IF signal type, the parameter value is changed, but it is not used by the interface module until the port configuration is changed to parallel or parallel interleaved, *and* the signal type is changed to IQ.

The query returns the currently set value, regardless of the port configuration, you must query all four states (clocks per sample, port configuration, data direction, and signal type) to know the interface module's current setup

#### Example

```
:DIG:CLOC:CPS 2
```

The preceding example sets two clock cycles for each sample.

**\*RST** 1

**Range** 1, 2, or 4

**Key Entry** **Clocks Per Sample**

## :DIGital:CLOCK:PHASe

Supported ES267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:CLOCK:PHASe <val>
```

```
[ :SOURce]:DIGital:CLOCK:PHASe?
```

This command sets the phase for the clock relative to the leading edge transition of the data. At 0 degrees the clock and leading edge of the data signal are aligned. Any phase value between 0 and 360 degrees can be used in the command, however, the signal generator rounds up or down to get 90, 180, 270 and 0 degree settings. For example 140 degrees will cause the signal generator to use the 180 degree setting.

If this command is executed when the clock rate is less than 10 MHz or greater than 200 MHz, the resolution of this setting changes to 180 degrees, and the maximum phase becomes 180 degrees.

### Example

```
:DIG:CLOC:PHAS 90DEG
```

The preceding example sets the clock phase to 90 degrees. The clock signal leading edge transition will be delayed by 1/4 of a clock period relative to the leading edge data transition.

**\*RST** +0.00000000E+000

**Range** 0 to 360 degrees

**Key Entry** **Clock Phase**

## :DIGital:CLOCK:POLarity

Supported ES267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:CLOCK:POLarity POSitive|NEGative
```

```
[ :SOURce]:DIGital:CLOCK:POLarity?
```

This command sets the alignment for the clock signal to positive or negative. Positive selects the leading edge transition of the clock signal to align with the leading edge data transition and negative selects the falling edge transition of the clock signal to align with the leading edge of the data.

### Example

```
:DIG:CLOC:POL NEG
```

The preceding example sets the clock falling edge transition to align with the leading edge data transition.

**\*RST** POS

**Key Entry** **Clock Polarity**

## :DIGital:CLOCK:RATE

Supported ES267D Option 601 or 602 with Option 003 or 004

```
[ :SOURCE]:DIGital:CLOCK:RATE <val>
[:SOURCE]:DIGital:CLOCK:RATE?
```

This command sets the clock rate. If an external clock is used, the rate set with this command must match the external clock rate. Only clock phase settings of 0 or 180 degrees are valid for a clock rate setting below 10 MHz or above 200 MHz. The variable <val> is expressed in hertz

### Example

```
:DIG:CLOC:RATE 200MHZ
```

The preceding example sets the clock rate to 200 megahertz.

**\*RST** +1.000000E+008  
**Range** 1 kHz to 400 MHz  
**Key Entry** **Clock Rate**

## :DIGital:CLOCK:REFerence:FREQuency

Supported ES267D Option 601 or 602 with Option 004

```
[ :SOURCE]:DIGital:CLOCK:REFerence:FREQuency <freq>
[:SOURCE]:DIG:CLOC:REF:FREQ?
```

This command allows you to specify the frequency of the external reference supplied to the Freq Ref connector. This command is valid only when the clock source is set to internal.

If this command is executed when the clock source is not set to internal, the parameter value is changed, but it is not used by the signal generator until the clock source is changed to internal.

Because a query returns the currently set value, regardless of the clock source, you must query both states (reference frequency and clock source) to know the signal generator's current setup.

### Example

```
:DIG:CLOC:REF:FREQ 50MHZ
```

The preceding example specifies a 50 megahertz external reference frequency.

**\*RST** +1.0000000E+007  
**Range** 1 to 100 MHz  
**Key Entry** **Reference Frequency**

## **:DIGital:CLOCK:SOURce**

Supported E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:CLOCK:SOURce INTernal|EXTernal|DEvice  
[:SOURce]:DIG:CLOC:SOURce?
```

This command selects one of three possible clock sources.

### **Example**

```
:DIG:CLOC:SOUR DEV
```

The preceding example uses the “Device Interface Connector” input clock.

**\*RST** INT

**Key Entry** Clock Source

## **:DIGital:CLOCK:SKEW**

Supported E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:CLOCK:SKEW <val>  
[:SOURce]:DIGital:CLOCK:SKEW?
```

This command sets the clock signal skew value. The skew is a fine-tune adjustment for the course tune clock phase function and helps to align the clock with valid data states. This is useful at high clock rates and available only for clock frequencies above 10 megahertz. The variable <val> is expressed in nanoseconds.

### **Example**

```
:DIG:CLOC:SKEW 2NS
```

The preceding example sets the clock skew to 2 nanoseconds.

**\*RST** +0.00000000E+000 ns

**Range** -5 ns to 5 ns

**Key Entry** Clock Skew

## **:DIGital:DATA:ALIGNment**

Supported E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:ALIGNment MSB|LSB  
[:SOURce]:DIGital:DATA:ALIGNment?
```

This command selects the bit alignment for word less than 16 bits in length. The MSB (most significant bit) selection maintains the MSB of the word on the same data line while the LSB (least significant bit) will move depending on the word size. The opposite effect occurs when the alignment is set to LSB.



**Example**

```
:DIG:DATA:ALIG MSB
```

The preceding example sets the MSB word format.

```
*RST          LSB
```

**Key Entry**            **Word Alignment**

**:DIGital:DATA:BORDER**

Supported            E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:BORDER MSB|LSB
[:SOURce]:DIGital:DATA:BORDER?
```

This command selects the bit order for data transmitted through the N5102A module. Data can be in least significant (LSB) bit first or most significant (MSB) bit first.

**Example**

```
:DIG:DATA:BORD MSB
```

The preceding example specifies data in MSB first format.

```
*RST          LSB
```

**Key Entry**            **Bit Order**

**:DIGital:DATA:DIRection**

Supported            E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:DIRection OUTPut|INPut
[:SOURce]:DIGital:DATA:DIRection?
```

This command selects an input or output direction for data flow through the N5102A module.

**Example**

```
:DIG:DATA:DIR INP
```

The preceding example selects input as the direction of data flow.

```
*RST          Output (unless only Option 004 is installed)
```

**Key Entry**            **Direction**

## :DIGital:DATA:IGain

Supported E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:IGain <val>  
[:SOURce]:DIGital:DATA:IGain?
```

This command adjust the gain of the I data in the N5102A module. The adjustment does not affect the Q data. The variable <val> is expressed as a percentage delta from 100%.

The offset is an adjustment to the analog level that is represented by the digital sample.

The analog voltage is limited to a 16-bit data sample. If the amplitude of the signal, after gain is applied, cannot be represented by 16 bits, the signal will be clipped.

### Example

```
:DIG:DATA:IG 10
```

The preceding example turns off wideband amplitude modulation.

**\*RST** +0.00000000E+000

**Range** -12.5 to 12.5

**Key Entry** I Gain

## :DIGital:DATA:INEGate

Supported E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:INEGate OFF|ON|0|1  
[:SOURce]:DIGital:DATA:INEGate?
```

This command enables or disables the negation of the I data sample. Negation changes the sample by expressing it in two's complement form, multiplying by negative one, and converting back to the selected numeric format. This can be done for I samples, Q samples, or both.

The sample or word represents a quantized analog voltage level. For a 16-bit sample, the range is from 0 to 65535 in offset binary or -32768 to + 32767 in 2's complement mode.

### Example

```
:DIG:DATA:INEG ON
```

The preceding example enables negation of the I data.

**\*RST** 0

**Key Entry** Negate I

## **:DIGital:DATA:IOFFset**

**Supported** E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:IOFFset <val>
[:SOURce]:DIGital:DATA:IOFFset?
```

This command adjusts the DC offset for I data. The command is available for the N5102A module output mode. The variable <val> is expressed as a +/- 100% of the full scale value. Refer to the *E8257D/67D, E8663D PSG Signal Generators Key Reference* for more information.

### **Example**

```
:DIG:DATA:IOFF 40
```

The preceding example sets the I offset to 40% of full scale.

**\*RST** +0.00000000E+000

**Range** -100 to +100

**Key Entry** **I Offset**

## **:DIGital:DATA:IQSWap**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:IQSWap OFF|ON|0|1
[:SOURce]:DIGital:DATA:IQSWap?
```

This command enables or disables swapping of the I and Q data. When enabled, the I data is sent to the N5102A's Q bus and the Q data is sent to the I bus.

### **Example**

```
:DIG:DATA:IQSW ON
```

The preceding example enables swapping of I and Q data.

**\*RST** 0

**Key Entry** **Swap IQ**

## **:DIGital:DATA:NFORmat**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:NFORmat OBINary|TCOMplement
[:SOURce]:DIGital:DATA:NFORmat?
```

This command selects the binary format used to represent the transmitted data values. The selections are offset binary or 2's complement.

### **Example**

```
:DIG:DATA:NFOR OBIN
```

The preceding example selects the offset binary format to represent data values.

**\*RST** TCOM

**Key Entry** **Numeric Format**

### **:DIGital:DATA:POLarity:FRAME**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:POLarity:FRAME POSitive|NEGative  
[:SOURce]:DIGital:DATA:POLarity:FRAME?
```

This command selects the polarity of the frame marker for serial transmission. The frame marker indicates the beginning of each sample or byte of data. The command is valid for serial transmission only.

**POS** This choice selects a positive polarity. The frame marker is high for the first data sample.

**NEG** This choice selects a negative polarity. The frame marker is low for the first data sample.

#### **Example**

```
:DIG:DATA:POL:FRAM NEG
```

The preceding example selects a negative polarity for the frame marker.

**\*RST** POS

**Key Entry** Frame Polarity

### **:DIGital:DATA:POLarity:IQ**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:POLarity:IQ POSitive|NEGative  
[:SOURce]:DIGital:DATA:POLarity:IQ?
```

This command selects the logic level for I and Q data. Positive selects a high logic level at the output as a digital one and negative selects a low logic level at the output as a digital one.

**POS** This choice selects a logic high level as digital one.

**NEG** This choice selects a logic low level as a digital one.

#### **Example**

```
:DIG:DATA:POL:IQ NEG
```

The preceding example turns off wideband amplitude modulation.

**\*RST** POS

**Key Entry** IQ Polarity

## :DIGital:DATA:QGain

**Supported** E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:QGain <val>
[:SOURce]:DIGital:DATA:QGain?
```

This command adjusts the gain for Q data in the N5102A module. The adjustment does not affect the I data. The variable <val> is expressed as a percentage delta from 100%. The offset is an adjustment to the analog level that is represented by the digital sample. The analog voltage is limited to a 16-bit data sample.

### Example

```
:DIG:DATA:QG 10
```

The preceding example increases the gain for Q data by 10% above the nominal value.

**\*RST** +0.00000000E+000

**Range** -12.5 to 12.5

**Key Entry** **Q Gain**

## :DIGital:DATA:QNEGate

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:QNEGate OFF|ON|0|1
[:SOURce]:DIGital:DATA:QNEGate?
```

This command enables or disables the negation of the Q data sample. Negation changes the sample by expressing it in two's complement form, multiplying by negative one, and converting back to the selected numeric format.

The sample or word represents a quantized analog voltage level. For a 16-bit sample, the range is from 0 to 65535 in offset binary or -32768 to + 32767 in 2's complement mode.

### Example

```
:DIG:DATA:QNEG ON
```

The preceding example enables negation of the Q data.

**\*RST** 0

**Key Entry** **Negate Q**

## **:DIGital:DATA:QOFFset**

**Supported** E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:QOFFset <val>  
[:SOURce]:DIGital:DATA:QOFFset?
```

This command adjusts the DC offset for Q data. The command is available for the N5102A module output mode. The variable <val> is expressed as a +/- 100% of the full scale value.

### **Example**

```
:DIG:DATA:QOFF 40
```

The preceding example sets the Q offset to 40% of full scale.

**\*RST** +0.00000000E+000

**Range** -100 to 100 percent

**Key Entry** **Q Offset**

## **:DIGital:DATA:ROTation**

**Supported** E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:ROTation <val>  
[:SOURce]:DIGital:DATA:ROTation?
```

This command rotates the IQ data in the IQ plane. This command is valid for the N5102A output mode. The variable <val> is expressed in degrees.

### **Example**

```
:DIG:DATA:ROT 45
```

The preceding example rotates the IQ constellation 45 degrees.

**\*RST** +1.00000000E+000

**Range** 0 to 360 degrees

**Key Entry** **Rotation**

## **:DIGital:DATA:SCALing**

**Supported** E8267D Option 601 or 602 with Option 003

```
[ :SOURce]:DIGital:DATA:SCALing <val>  
[:SOURce]:DIGital:DATA:SCALing?
```

This command enables scaling of the I and Q data to the level indicated by the <val> variable. This command is valid for the N5102A output mode. The variable <val> is expressed as a percentage.

**Example**

```
:DIG:DATA:SCAL 50
```

The preceding example scales the I and Q data to amplitude to 50% of the nominal value.

**\*RST** +1.00000000E+002

**Range** -100 to 100 percent

**Key Entry** **Scaling**

**:DIGital:DATA:SIZE**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[[:SOURce]:DIGital:DATA:SIZE <val>
```

```
[[:SOURce]:DIGital:DATA:SIZE?
```

This command selects the number of bits in each sample. A sample can have a maximum word length of 16 bits.

**Example**

```
:DIG:DATA:SIZE 8
```

The preceding example sets the sample word size to eight bits.

**\*RST** +1.600000000E+001

**Range** 4 to 16

**Key Entry** **Word Size**

**:DIGital:DATA:STYPe**

**Supported** E8267D Option 601 or 602 with Option 003

```
[[:SOURce]:DIGital:DATA:STYPe IQ|IF
```

```
[[:SOURce]:DIGital:DATA:STYPe?
```

This command selects the output format for the IQ data. The IQ selection outputs digital I and Q data. Whereas the IF (intermediate frequency) selection modulates the I and Q data onto the IF frequency. The IF is calculated as 1/4 the clock sample rate. This command is valid only for the N5102A output mode.

**IQ** This choice outputs I and Q digital data.

**IF** This choice outputs a modulated signal.

**Example**

```
:DIG:DATA:STYP IF
```

The preceding example sets the output data to IF.

**\*RST** IQ

**Key Entry** **Signal Type**

## :DIGital:DATA:TYPE

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DATA:TYPE SAMPles|PFSamples  
[:SOURce]:DIGital:DATA:TYPE?
```

This command selects filtered baseband data or unfiltered baseband data as the transmitted data type.

If this command is executed while an ARB modulation format is active, the parameter choice is changed, but it is not *used* by the interface module until a real-time modulation format is turned on.

Because a query returns the current choice, regardless of whether or not an ARB format is active, you must query both states (data type and the modulation format) to know the signal generator's current setup.

**SAMPles** This choice selects DAC samples at the data transmitted.

**PFSamples** This choice selects pre-filtered samples which are unfiltered I and Q data.

### Example

```
:DIG:DATA:TYPE PFS
```

The preceding example sets the data type to pre-filtered I and Q data.

**\*RST** SAMP

**Key Entry** Data Type

## :DIGital:DIAGnostic:LOOPback

**Supported** E8267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:DIAGnostic:LOOPback DIGBus|CABLE|N5102A|DEVIce  
[:SOURce]:DIGital:DIAGnostic:LOOPback?
```

This command selects a loop back test that validates the integrity of digital data. Refer to the *E8257D/67D, E8663D PSG Signal Generators Key Reference* for more information.

**DIGBus** This choice selects a loop back test using the Digital Bus Loop Back Fixture test board.

**CABLE** This choice selects a loop back test on the PSG Digital Bus connector at the signal generator side.

**N5102A** This choice selects a loop back test for the N5102A module.

**DEVIce** This choice selects a loop back test using the LOOP BACK TEST SINGLE ENDED IO DUAL 40 PIN board.

### Example

```
:DIG:DIAG:LOOP?
```

The preceding example runs the diagnostic test for device and returns a pass or fail state.

**\*RST** Device Intfc

**Key Entry** Loop Back Test Type



## :DIGital:LOGic[:TYPE]

**Supported** ES267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:LOGic[:TYPE] LVDS|LVTT1|CMOS15|CMOS18|CMOS25|CMOS33
[:SOURce]:DIGital:LOGic[:TYPE]?
```

This command selects the logic data type used by the device being tested.

LVDS	This choice selects low voltage differential signaling as the logic data type.
LVTT1	This choice selects a low voltage TTL signal as the logic data type.
CMOS15	This choice selects a 1.5 volt CMOS signal as the logic data type.
CMOS18	This choice selects a 1.8 volt CMOS signal as the logic data type.
CMOS25	This choice selects a 2.5 volt CMOS signal as the logic data type.
CMOS33	This choice selects a 3.3 volt CMOS signal as the logic data type.

### Example

```
:DIG:LOG CMOS15
```

The preceding example selects 1.5 volt CMOS as the logic data type.

```
*RST CMOS33
```

**Key Entry**            **Logic Type**

## :DIGital:PCONfig

**Supported** ES267D Option 601 or 602 with Option 003 or 004

```
[ :SOURce]:DIGital:PCONfig PARallel|SERial|PINTIQ|PINTI
[:SOURce]:DIGital:PCONfig?
```

This command selects the data transmission type used for communication between the N5102A module and the device under test. Refer to the *E8257D/67D, E8663D PSG Signal Generators Key Reference* for more information.

PARallel	This choice selects parallel data transmission.
SERial	This choice selects serial data transmission.
PINTIQ	This choice selects parallel interleaving data transmission. The I data is transmitted on the rising clock edge and the Q data on the falling edge.
PINTI	This choice selects parallel interleaving data transmission. The Q data is transmitted on the rising clock edge and the I data on the falling edge.

### Example

```
:DIG:PCON PINTI
```

The preceding example selects parallel interleaving using the QI format

```
*RST PAR
```

**Key Entry**            **Port Config**

## **:DIGital:PRESet:PTHROUGH**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

[[:SOURce]:DIGital:PRESet:PTHROUGH

This command sets up the preset condition for the N5102A module and allows transmission of data through the module with no modifications. The command is valid only when a modulation format is active.

### **Example**

```
:DIG:PRES:PTHR
```

The preceding example sets the N5102A module to a preset condition and allows data to pass through unmodified.

**Key Entry**            **Pass Through Preset**

## **:DIGital[:STATe]**

**Supported** E8267D Option 601 or 602 with Option 003 or 004

[[:SOURce]:DIGital[:STATe] 0|1|OFF|ON

[[:SOURce]:DIGital[:STATe]?

This command enables or disables the operating state of the N5102A module.

### **Example**

```
:DIG ON
```

The preceding example turns on the N5102A module.

**\*RST**                OFF

**Key Entry**            **N5102A Off On**

---

# 7 SCPI Command Compatibility

This chapter provides a compatibility listing of SCPI commands. Many commands unique to other Agilent signal generator models are also supported by the PSG Signal Generator:

- “:SYSTem:IDN” on page 347
- “E8257D/67D Compatible Commands” on page 348
- “E8241A/44A/51A/54A and the E8247C/57C/67C PSG Compatible SCPI Commands” on page 349
- “8340B/41B and 8757D Compatible Commands” on page 350
- “836xxB/L Compatible SCPI Commands” on page 363
- “8373xB and 8371xB Compatible SCPI Commands” on page 379
- “8375xB Compatible SCPI Commands” on page 387
- “8662A/63A Compatible Commands” on page 399

## :SYSTem:IDN

Supported            All

:SYSTem:IDN "<string>"

This command modifies the identification string that the \*IDN? query returns. Sending an empty string returns the query output to its factory shipped setting. The maximum string length is 72 characters.

Modification of the \*IDN? query output enables the PSG to identify itself as another signal generator when it is used as a backward compatible replacement.

The display diagnostic information, shown by pressing the **Diagnostic Info** softkey, is not affected by this command.

### Example

```
:SYST:IDN "Agilent Technologies, Exxxx, US4000000, c.00.00.1234"
```

The preceding example changes and sets the identification string for the signal generator.

## E8257D/67D Compatible Commands

The following commands are compatible with the E8257D, E8267D, and E8663D signal generators. These commands were documented in earlier versions of firmware but are now deprecated and may be removed from future firmware versions.

### :DATA:PRAM?

**Supported** E8267D with Option 601 or 602

:MEMory:DATA:PRAM?

This query determines whether there is a user-defined pattern in the pattern RAM (PRAM). This command is not compatible with the [":DATA:PRAM:FILE:BLOCK"](#) or [":DATA:PRAM:FILE:LIST"](#) commands.

\*RST 0

### :DATA:PRAM:BLOCK

**Supported** E8267D with Option 601 or 602

:MEMory:DATA:PRAM:BLOCK <data\_block>

This command downloads the block-formatted data directly into pattern RAM. This command is still valid for backward compatibility; however, it has been replaced by the [":DATA:PRAM:FILE:BLOCK"](#) command.

### :DATA:PRAM:LIST

**Supported** E8267D Option 601 or 602

:MEMory:DATA:PRAM:LIST <uint8>[, <uint8>, <...>]

This command downloads the list-formatted data directly into pattern RAM. This command is still valid for backward compatibility; however, it has been replaced by the [":DATA:PRAM:FILE:LIST"](#) command.

<uint8> This variable is any of the valid 8-bit, unsigned integer values between 0 and 255.

[, <uint8>, <...>] This variable identifies the value of the second and subsequent 8-bit unsigned integer variables.

**Range** 0 to 255

### :DELeTe:WFM1

**Supported** E8267D with Option 601 or 602

:MMEMory:DELeTe:WFM1

This command clears the memory file system of all volatile arbitrary waveform (WFM1) files. It performs the same function as DELeTe:WFM command. Refer to [":DELeTe:WFM"](#) on page 61.

**Key Entry** Delete All WFM1 Files

## **E8241A/44A/51A/54A and the E8247C/57C/67C PSG Compatible SCPI Commands**

All commands are fully supported. To use the commands, select **SCPI** as the remote language. See “:LANGuage” on page 93 for selecting the language type.

## 8340B/41B and 8757D Compatible Commands

The tables in this section provide the following:

**Table 7-1 on page 350:** a comprehensive list of 8340B/41B and 8757D programming codes, listed in alphabetical order. The equivalent SCPI command sequence for each supported code is provided; codes that are *not* supported by the PSG family are indicated as such in the command column.

**Table 7-2 on page 361:** a list of the implemented 8340B/41B and 8757D programming codes that set the active function. This table also indicates which codes are compatible with the RB command (knob), and lists the operation active (OA) query, the operation prior (OP) query, and the increment (up), and the decrement (down) SCPI commands.

---

**NOTE** Compatibility is provided for GPIB only; RS-232 and LAN are *not* supported.

---

When using the programming codes in this section, you can:

- set the PSG system language to 8340 or 8757 for the current session:

**Utility > GPIB/RS-232 LAN > Remote Language > 8340B (or 8757D)**

or send the command:

```
:SYST:LANG "8340" (or "8757")
```

- set the PSG system language to 8340 or 8757 so that it does not reset with either preset, instrument power cycle or \*RST command:

**Utility > Power On/Preset > Remote Language > 8340B (or 8757D)**

or send the command:

```
:SYST:PRESET:LANG "8340" (or "8757")
```

- set the \*IDN? response to any 8340-like response you prefer. Refer to the [:SYSTEM:IDN](#) command on [page 347](#).

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
A1	Internal leveling mode	Y	Y	<code>[:SOURCE]:POWER:ALC:SOURCE INTERNAL</code>
A2	External leveling mode with diode detector	Y	Y	<code>[:SOURCE]:POWER:ALC:SOURCE DIODE</code> <code>[:SOURCE]:POWER:ALC:SOURCE:EXTERNAL:COUPLING &lt;val&gt; dB</code>
A3	External leveling mode with power meter	Y	Y	<i>supported, but has no effect on PSG</i>
AK0	Amplitude markers off	Y	Y	<code>[:SOURCE]:MARKER:AMPLITUDE OFF 0</code>
AK1	Amplitude markers on	Y	Y	<code>[:SOURCE]:MARKER:AMPLITUDE ON 1</code>
AL0	Alternate sweep mode off	Y	Y	<code>:SYSTEM:ALTERNATE:STATE OFF</code>
AL1	Alternate sweep mode on	Y	Y	<code>:SYSTEM:ALTERNATE:STATE ON</code> <code>:SYSTEM:ALTERNATE n</code>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
AM0	Amplitude modulation off	Y	N	<code>[ :SOURCE]:AM1:STATe OFF 0</code> <code>[ :SOURCE]:AM2:STATe OFF 0</code>
AM1	Amplitude modulation on	Y	N	<code>[ :SOURCE]:AM1:STATe OFF 0</code> <code>[ :SOURCE]:AM2:SOURce EXT 1</code> <code>[ :SOURCE]:AM2:EXTErnal[1]:COUPling DC</code> <code>[ :SOURCE]:AM2:DEPTh 100</code> <code>[ :SOURCE]:AM2:EXTErnal[1]:IMPedance 600</code> <code>[ :SOURCE]:AM2:STATe ON 1</code>
AS0	Alternate state selection: select current front panel	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
AS1	Alternate state selection: select recalled state	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
AT	Set attenuator	Y	N	<code>[ :SOURCE]:POWEr:ATTenuation &lt;val&gt;&lt;unit&gt;</code>
AU	Auto-coupled mode to obtain shortest possible sweep time	Y	N	<code>[ :SOURCE]:SWEep:TIME:AUTO ON 1</code>
BC	Advance to next frequency bandcrossing	N	N	<i>not supported</i>
C1	1 MHz crystal marker frequency	N	Y	<i>supported, but has no effect on PSG</i>
C2	10 MHz crystal marker frequency	N	Y	<i>supported, but has no effect on PSG</i>
C3	50 MHz crystal marker frequency	N	Y	<i>supported, but has no effect on PSG</i>
C4	External crystal marker frequency	N	Y	<i>supported, but has no effect on PSG</i>
CA0	Amplitude crystal markers off	N	Y	<i>supported, but has no effect on PSG</i>
CA1	Amplitude crystal markers on	N	Y	<i>supported, but has no effect on PSG</i>
CF	Center frequency (step sweep)	Y	Y	<code>[ :SOURCE]:SWEep:MODE AUTO</code> <code>[ :SOURCE]:FREQuency:MODE SWEep</code> <code>[ :SOURCE]:FREQuency:CENTer &lt;val&gt;&lt;unit&gt;</code>
CL0	Intensity crystal markers off	N	Y	<i>supported, but has no effect on PSG</i>
CL1	Intensity crystal markers on	N	Y	<i>supported, but has no effect on PSG</i>
CS	Clear both status bytes	Y	Y	*CLS
CW	Set CW frequency	Y	Y	<code>[ :SOURCE]:SWEep:MODE AUTO</code> <code>[ :SOURCE]:FREQuency:MODE CW</code> <code>[ :SOURCE]:FREQuency[:CW] &lt;val&gt;&lt;unit&gt;</code>
DB	dB(m) terminator	Y	Y	DB
DF	Delta frequency (step sweep)	Y	Y	<code>[ :SOURCE]:SWEep:MODE AUTO</code> <code>[ :SOURCE]:FREQuency:MODE SWEep</code> <code>[ :SOURCE]:FREQuency:SPAN &lt;val&gt; &lt;unit&gt;</code>
DM	dB(m) terminator	Y	Y	DB
DN	Step down (decrements active function by step value)	Y	Y	<i>supported, see Table 6-2 on page 234</i>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
DP0	Display blanking off	N	Y	DISPlay[:WINDow][:STATe] ON 1
DP1	Display blanking on	N	Y	DISPlay[:WINDow][:STATe] OFF 0
DU0	Display update off	Y	Y	DISPlay[:WINDow][:STATe] OFF 0
DU1	Display update on	Y	Y	DISPlay[:WINDow][:STATe] ON 1
EF	Entry display off	Y	Y	DISPlay[:WINDow][:STATe] ON 1
EK	Enable knob	N	N	<i>not supported</i>
EM0	Extended marker mode off	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
EM1	Extended marker mode on	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
F1	20 MHz/V FM sensitivity	N	N	<i>not supported</i>
F2	6 MHz/V FM sensitivity	N	N	<i>not supported</i>
FA	Start frequency (step sweep)	Y	Y	[:SOURce]:SWEep:MODE AUTO [:SOURce]:FREQuency:MODE SWEep [:SOURce]:FREQuency:START <val><unit>
FB	Stop frequency (step sweep)	Y	Y	[:SOURce]:SWEep:MODE AUTO [:SOURce]:FREQuency:MODE SWEep [:SOURce]:FREQuency:STOP <val><unit>
FL0	CW filter off	N	Y	<i>supported, but has no effect on PSG</i>
FL1	CW filter on	N	Y	<i>supported, but has no effect on PSG</i>
FM0	Frequency modulation off	Y	N	[:SOURce]:FM1:STATe OFF 0 [:SOURce]:FM2:STATe OFF 0
FM1	Frequency modulation on	Y	N	[:SOURce]:FM1:STATe OFF 0 [:SOURce]:FM2:SOURce EXT2 [:SOURce]:FM2:EXTErnal2:COUpling DC [:SOURce]:FM2:EXTErnal2:IMPedance 50 [:SOURce]:FM2:STATe ON 1
FM1	Frequency modulation sensitivity	Y	N	[:SOURce]:FM2[:DEVIation] <val><unit>
FP	Fast phaselock	Y	N	<i>supported, but has no effect on PSG</i>
GZ	GHz terminator	Y	Y	GHZ
HZ	Hz terminator	Y	Y	HZ
IF	Increment frequency	Y	N	TRIGger[:SEQuence][:IMMediate] or [:SOURce]:FREQuency[:CW] UP
IL	Input learn string	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>



**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
IP	Instrument preset	Y	N	<pre> SYSTEM:PRESet [:SOURce]:FREQuency[:CW]:STEP [:INCRement] 1 GHZ  [:SOURce]:FREQuency:MULTiplier &lt;saved multiplier&gt;  [:SOURce]:SWEep:MODE AUTO [:SOURce]:FREQuency:MODE SWEep [:SOURce]:FREQuency:START 2 GHz or MIN [:SOURce]:FREQuency:STOP MAX [:SOURce]:POWer[:LEVel][:IMMediate] [:AMPLitude] 0 dB  OUTput[:STATe] ON 1 </pre>
IP	Instrument preset	N	Y	<pre> SYSTEM:PRESet SYSTEM:LANGUage "8757" [:SOURce]:SWEep:MODE AUTO [:SOURce]:FREQuency:MODE SWEep [:SOURce]:FREQuency:START 2 GHz or MIN [:SOURce]:FREQuency:STOP MAX [:SOURce]:POWer[:LEVel][:IMMediate] [:AMPLitude] 0 dB  OUTput[:STATe] ON 1 </pre>
IX	Input micro learn string	N	Y	<i>supported, but has no effect on PSG</i>
KR	Key release	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
KZ	kHz terminator	Y	Y	KHZ
M0 MO	Frequency marker off	Y	Y	[:SOURce]:MARKer[n]:[STATe] OFF 0
MA	Turn on and set frequency marker 0	Y	Y	[:SOURce]:MARKer0:[STATe] ON 1 [:SOURce]:MARKer0:FREQuency <val><unit>
M1	Turn on and set frequency marker 1	Y	Y	[:SOURce]:MARKer1:[STATe] ON 1 [:SOURce]:MARKer1:FREQuency <val><unit>
M2	Turn on and set frequency marker 2	Y	Y	[:SOURce]:MARKer2:[STATe] ON 1 [:SOURce]:MARKer2:FREQuency <val><unit>
M3	Turn on and set frequency marker 3	Y	Y	[:SOURce]:MARKer3:[STATe] ON 1 [:SOURce]:MARKer3:FREQuency <val><unit>
M4	Turn on and set frequency marker 4	Y	Y	[:SOURce]:MARKer4:[STATe] ON 1 [:SOURce]:MARKer4:FREQuency <val><unit>
M5	Turn on and set frequency marker 5	Y	Y	[:SOURce]:MARKer5:[STATe] ON 1 [:SOURce]:MARKer5:FREQuency <val><unit>
M6	Turn on and set frequency marker 6	Y	Y	[:SOURce]:MARKer6:[STATe] ON 1 [:SOURce]:MARKer6:FREQuency <val><unit>
M7	Turn on and set frequency marker 7	Y	Y	[:SOURce]:MARKer7:[STATe] ON 1 [:SOURce]:MARKer7:FREQuency <val><unit>
M8	Turn on and set frequency marker 8	Y	Y	[:SOURce]:MARKer8:[STATe] ON 1 [:SOURce]:MARKer8:FREQuency <val><unit>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
M9	Turn on and set frequency marker 9	Y	Y	[ :SOURCE]:MARKer9:[STATe] ON 1 [ :SOURCE]:MARKer9:FREQuency <val><unit>
MC	Active marker to center frequency	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
MD	Marker delta	N	N	<i>not supported</i>
MP0	Marker 1-2 sweep off	N	N	<i>not supported</i>
MP1	Marker 1-2 sweep on	N	N	<i>not supported</i>
MS	Milliseconds terminator	Y	Y	MS
MZ	MHz terminator	Y	Y	MHZ
NA	Network analyzer mode	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
NT	Network analyzer trigger	N	Y	<i>supported, but has no effect on PSG</i>
OA	Output active parameter	Y	Y	<i>supported, see Table 6-2 on page 234</i>
OB	Output next bandcross frequency	N	N	<i>not supported</i>
OC	Output coupled parameters (start frequency, center frequency, sweep time)	Y	Y	[ :SOURCE]:FREQuency:START? [ :SOURCE]:FREQuency:CENTer? [ :SOURCE]:SWEep:TIME?
OD	Output diagnostic values	N	N	<i>not supported</i>
OE	Output when executed	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
OF	Output fault	Y	N	<i>supported, but no equivalent SCPI command sequence</i>
OI	Output identification	Y	Y	*IDN?
OK	Output last lock frequency	N	N	<i>not supported</i>
OL	Output learn string	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
OM	Output mode string	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
OP	Output interrogated parameter	Y	Y	<i>supported, see Table 6-2 on page 234</i>
OPA2	Output external detector coupling factor	Y	Y	[ :SOURCE]:POWer:ALC:SOURCE:EXTernal:COUPLing?
OPAT	Output attenuator	Y	N	[ :SOURCE]:POWer:ATTenuation?
OPCF	Output center frequency	Y	Y	[ :SOURCE]:FREQuency:CENTer?
OPCW	Output CW frequency	Y	Y	[ :SOURCE]:FREQuency:CW?
OPDF	Output delta frequency	Y	Y	[ :SOURCE]:FREQuency:SPAN?
OPFA	Output start frequency	Y	Y	[ :SOURCE]:FREQuency:START?
OPFB	Output stop frequency	Y	Y	[ :SOURCE]:FREQuency:STOP?
OPFM1	Output FM sensitivity	Y	N	[ :SOURCE]:FM2[:DEVIation]?

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
OPMA	Output marker 0 frequency	Y	Y	[ :SOURCE ] : MARKer0 : FREQuency?
OPM1	Output marker 1 frequency	Y	Y	[ :SOURCE ] : MARKer1 : FREQuency?
OPM2	Output marker 2 frequency	Y	Y	[ :SOURCE ] : MARKer2 : FREQuency?
OPM3	Output marker 3 frequency	Y	Y	[ :SOURCE ] : MARKer3 : FREQuency?
OPM4	Output marker 4 frequency	Y	Y	[ :SOURCE ] : MARKer4 : FREQuency?
OPM5	Output marker 5 frequency	Y	Y	[ :SOURCE ] : MARKer5 : FREQuency?
OPM6	Output marker 6 frequency	Y	Y	[ :SOURCE ] : MARKer6 : FREQuency?
OPM7	Output marker 7 frequency	Y	Y	[ :SOURCE ] : MARKer7 : FREQuency?
OPM8	Output marker 8 frequency	Y	Y	[ :SOURCE ] : MARKer8 : FREQuency?
OPM9	Output marker 9 frequency	Y	Y	[ :SOURCE ] : MARKer9 : FREQuency?
OPPL	Output power level	Y	Y	[ :SOURCE ] : POWer [ : LEVel ] [ : IMMEDIATE ] [ : AMPLitude ] ?
OPPS	Output power sweep span	Y	Y	[ :SOURCE ] : POWer : SPAN?
OPSB	Output # of sweep buckets	N	N	<i>supported, but no equivalent SCPI command sequence</i>
OPSF	Output frequency step size	Y	Y	[ :SOURCE ] : FREQuency [ : CW ] : STEP [ : INCRement ] ?
OPSHA1	Output power level	Y	N	[ :SOURCE ] : POWer [ : LEVel ] [ : IMMEDIATE ] [ : AMPLitude ] ?
OPSHA2	Output ALC level	Y	N	[ :SOURCE ] : POWer : ALC : LEVel?
OPSHA3	Output ALC level	Y	N	[ :SOURCE ] : POWer : ALC : LEVel?
OPSHAZ	Output ALC level	Y	N	[ :SOURCE ] : POWer : ALC : LEVel?
OPSHCF	Output frequency step size	Y	N	[ :SOURCE ] : FREQuency [ : CW ] : STEP [ : INCRement ] ?
OPSHCW	Output swept CW frequency	Y	Y	[ :SOURCE ] : FREQuency : START? or [ :SOURCE ] : FREQuency : STOP?
OPSHFA	Output frequency multiplier	Y	Y	[ :SOURCE ] : FREQuency : MULTiplier?
OPSHFB	Output frequency offset	Y	Y	[ :SOURCE ] : FREQuency : OFFSet?
OPSHPL	Output power step size	Y	N	[ :SOURCE ] : POWer [ : LEVel ] [ : IMMEDIATE ] [ : AMPLitude ] : STEP [ : INCRement ] ?
OPSHPS	Output ALC level	Y	Y	[ :SOURCE ] : POWer : ALC : LEVel?
OPSHRF	Output power level	Y	N	[ :SOURCE ] : POWer [ : LEVel ] [ : IMMEDIATE ] [ : AMPLitude ] ?
OPSHSL	Output attenuator	Y	N	[ :SOURCE ] : POWer : ATTenuation?
OPSHSN	Output sweep step points	N	Y	[ :SOURCE ] : SWEEp : POINTs?
OPSL	Output power slope	Y	Y	[ :SOURCE ] : POWer : SLOPe?

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
OPSM	Output manual frequency	Y	Y	<code>[:SOURCE]:FREQUENCY:MANUAL?</code>
OPSN	Output sweep step points	Y	Y	<code>[:SOURCE]:SWEep:POINTS?</code>
OPSP	Output power step size	Y	Y	<code>[:SOURCE]:POWER[:LEVEL][:IMMEDIATE] [:AMPLITUDE]:STEP[:INCREMENT]?</code>
OPST	Output sweep time	Y	Y	<code>[:SOURCE]:SWEep:TIME?</code>
OPTL	Output sweep time limit	Y	Y	<code>[:SOURCE]:SWEep:TIME:LLIMIT?</code>
OR	Output internally measured power level	N	N	<i>not supported</i>
OS	Output status bytes	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
OX	Output micro learn string	N	Y	<i>supported, but has no effect on PSG</i>
PL	Set power level	Y	Y	<code>[:SOURCE]:POWER:ATTenuation:AUTO ON 1 [:SOURCE]:POWER[:LEVEL][:IMMEDIATE] [:AMPLITUDE] &lt;val&gt;&lt;unit&gt;</code>
PM0	Pulse modulation off	Y	Y	<code>[:SOURCE]:PULM:STATe OFF 0</code>
PM1	Pulse modulation on	Y	N	<code>[:SOURCE]:PULM:SOURce EXTERNAL [:SOURCE]:PULM:STATe ON 1</code>
PM1	27.8 KHz square wave pulse modulation on	N	Y	<code>[:SOURCE]:PULM:SOURce SCALAr [:SOURCE]:PULM:STATe ON 1</code>
PS0	Power sweep off	Y	Y	<code>[:SOURCE]:POWER:MODE FIXEd</code>
PS1	Power sweep on	Y	Y	<code>[:SOURCE]:POWER:MODE SWEep [:SOURCE]:POWER:SPAN &lt;val&gt; dB</code>
R2	Extended status byte #2 mask	N	Y	<i>supported, but has no effect on PSG</i>
RB	Control knob remotely	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
RC	Recall state	Y	Y	<code>*RCL &lt;reg_num&gt;[,&lt;seq_num&gt;]</code>
RE	Extended status byte mask	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
RF0	RF output off	Y	Y	<code>OUTPut[:STATe] OFF 0</code>
RF1	RF output on	Y	Y	<code>OUTPut[:STATe] ON 1</code>
RM	Status byte mask	Y	Y	<code>*SRE &lt;mask&gt;</code>
RP0	RF peaking off	Y	N	<i>supported, but has no effect on PSG</i>
RP0	RF blanking off	N	Y	<i>supported, but has no effect on PSG</i>
RP1	RF peaking on	Y	N	<i>supported, but has no effect on PSG</i>
RP1	RF blanking on	N	Y	<i>supported, but has no effect on PSG</i>
RS	Reset sweep	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
S1	Continuous sweep mode	Y	Y	[:SOURCE]:SWEep:MODE AUTO  [:SOURCE]:SWEep:GENeration ANALog :TRIGger[:SEQuence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] OFF
S2	Single sweep mode	Y	Y	[:SOURCE]:SWEep:MODE AUTO  [:SOURCE]:SWEep:GENeration ANALog :TRIGger[:SEQuence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] OFF
S3	Manual frequency sweep mode	Y	Y	[:SOURCE]:SWEep:MODE MANual  [:SOURCE]:SWEep:GENeration ANALog :TRIGger[:SEQuence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] OFF
SB	Number of sweep buckets	N	Y	<i>supported, but no equivalent SCPI command sequence</i>
SC	Seconds terminator	Y	Y	S
SF	Frequency step size	Y	Y	[:SOURCE]:FREQuency[:CW]:STEP[:INCRement] <val><unit>
SG	Single sweep mode	Y	Y	[:SOURCE]:SWEep:MODE AUTO  [:SOURCE]:SWEep:GENeration ANALog :TRIGger[:SEQuence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] OFF
SH	Shift prefix	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SH01	Blank display	N	Y	DISPlay[:WINDow][:STATe] OFF 0
SHA1	Disable ALC and set power level	Y	N	[:SOURCE]:POWer:ALC[:STATe] OFF 0  [:SOURCE]:POWer[:LEVel][:IMMEDIATE] [:AMPLitude] <val><unit>
SHA2	External leveling mode with millimeter head module	Y	N	[:SOURCE]:POWer:ALC:SOURce MMHead [:SOURCE]:POWer:ALC:LEVel <val>dB
SHA3	Directly control linear modulator circuit (bypassing ALC)	Y	N	[:SOURCE]:POWer:ATTenuation:AUTO OFF 0 [:SOURCE]:POWer:ALC[:STATe] OFF 0 [:SOURCE]:POWer:ALC:LEVel <val>dB
SHAK	Immediate YTF peak	Y	N	<i>supported, but has no effect on PSG</i>
SHAL	Retain multiplication factor on power on/off and preset	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SHAM	Pulse modulation enhancement	Y	N	<i>supported, but has no effect on PSG</i>
SHAZ	External leveling mode with millimeter head module	Y	N	[:SOURCE]:POWer:ALC:SOURce MMHead [:SOURCE]:POWer:ALC:LEVel <val>dB
SHCF	Frequency step size	Y	N	[:SOURCE]:FREQuency[:CW]:STEP[:INCRement] <val><unit>
SHCF	Coarse CW resolution	N	Y	<i>supported, but has no effect on PSG</i>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SHCW	Swept CW	N	Y	<code>[:SOURCE]:SWEep:MODE AUTO</code> <code>[:SOURCE]:FREQuency:MODE SWEep</code> <code>[:SOURCE]:FREQuency:START &lt;val&gt;&lt;unit&gt;</code> <code>[:SOURCE]:FREQuency:STOP &lt;val&gt;&lt;unit&gt;</code>
SHDF	Fine CW resolution	N	Y	<i>supported, but has no effect on PSG</i>
SHEF	Restore cal. const. access function	N	N	<i>not supported</i>
SHFA	Frequency multiplier	Y	Y	<code>[:SOURCE]:FREQuency:MULTiplier &lt;val&gt;</code>
SHFB	Frequency offset	Y	Y	<code>[:SOURCE]:FREQuency:OFFSet &lt;val&gt;&lt;unit&gt;</code>
SHIP	Reset multiplication factor to 1 and preset instrument	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SHM0	All frequency markers off	Y	Y	<code>[:SOURCE]:MARKer:AOFF</code>
SHM1	Turn on and set marker delta	N	Y	<code>[:SOURCE]:MARKer:MODE DELTa</code>
SHM2	Enable counter interface	N	Y	<i>supported, but has no effect on PSG</i>
SHM3	Disable counter interface	N	Y	<i>supported, but has no effect on PSG</i>
SHM4	Diagnostics: test/display results	N	N	<i>not supported</i>
SHM0	All frequency markers off	N	Y	<code>[:SOURCE]:MARKer:AOFF</code>
SHMP	Set start frequency to marker 1 and set stop frequency to marker 2	Y	Y	<code>[:SOURCE]:SWEep:MARKer:XFER</code>
SHPL	Power step size	Y	N	<code>[:SOURCE]:POWer[:LEVel][:IMMEDIATE]</code> <code>[:AMPLitude]:STEP[:INCRement] &lt;val&gt;</code>
SHPM	27.8 KHz square wave pulse modulation on	Y	Y	<code>[:SOURCE]:PULM:SOURce SCALar</code>  <code>[:SOURCE]:PULM:STATe ON 1:OUTPut</code> <code>:MODulation[:STATe] ON 1</code>
SHPS	Decouple attenuator and ALC (control ALC independently)	Y	Y	<code>[:SOURCE]:POWer:ATTenuation:AUTO OFF 0</code> <code>[:SOURCE]:POWer:ALC[:STATe] ON 1</code> <code>[:SOURCE]:POWer:ALC:LEVel &lt;val&gt;dB</code>
SHRC	Unlock save/recall	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SHRF	Disable ALC and set power level	Y	N	<code>[:SOURCE]:POWer:ALC[:STATe] OFF 0</code> <code>[:SOURCE]:POWer[:LEVel][:IMMEDIATE]</code> <code>[:AMPLitude] &lt;val&gt;&lt;unit&gt;</code>
SHRP	Auto track	Y	N	<i>supported, but has no effect on PSG</i>
SHS10	Disable display update	Y	N	<code>DISPlay[:WINDow][:STATe] OFF 0</code>
SHS11	Re-enable display update	Y	N	<code>DISPlay[:WINDow][:STATe] ON 1</code>
SHS3	Display fault diagnostic	N	N	<i>not supported</i>
SHSL	Set attenuator from front panel	Y	Y	<code>[:SOURCE]:POWer:ATTenuation &lt;val&gt;&lt;unit&gt;</code>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SHSN	Stepped sweep	N	Y	<code>[[:SOURCE]:SWEep:MODE AUTO [:SOURCE]:SWEep:GENeration STEPped [:SOURCE]:LIST:TYPE STEP  [:SOURCE]:LIST:TRIGger:SOURce IMMEDIATE :TRIGger[:SEQUence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] ON  [:SOURCE]:SWEep:POINts &lt;val&gt;</code>
SHSS	Reset step sizes to default values	N	Y	<i>supported, but has no effect on PSG</i>
SHST	Zoom function	N	N	<i>not supported</i>
SHSV	Lock save/recall	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SHT1	Test displays	N	N	<i>not supported</i>
SHT2	Bandcrossing penlift	N	N	<i>not supported</i>
SHT3	Display unlock indicators	N	N	<i>not supported</i>
SHGZ	IO Channel	N	N	<i>not supported</i>
SHMZ	IO Subchannel	N	N	<i>not supported</i>
SHKZ	Write to IO	N	N	<i>not supported</i>
SHHZ	Read from IO	N	N	<i>not supported</i>
SHVR	Frequency offset	N	N	<i>not supported</i>
SL0	Power slope off	Y	Y	<code>[[:SOURCE]:POWer:SLOPe:STATe OFF 0</code>
SL1	Power slope on	Y	N	<code>[[:SOURCE]:POWer:SLOPe:STATe ON 1 [:SOURCE]:POWer:SLOPe &lt;value&gt; [DB/GHz]</code>
SL1	Power slope on	N	Y	<code>[[:SOURCE]:POWer:SLOPe:STATe ON 1 [:SOURCE]:POWer:SLOPe &lt;value&gt; [DB/Hz]</code>
SM	Manual frequency sweep mode	Y	Y	<code>[[:SOURCE]:SWEep:MODE MANUal [:SOURCE]:FREQuency:MANUal &lt;val&gt;&lt;unit&gt;</code>
SN	Number of points in a stepped sweep	Y	Y	<code>[[:SOURCE]:SWEep:MODE AUTO [:SOURCE]:SWEep:GENeration STEPped [:SOURCE]:LIST:TYPE STEP  [:SOURCE]:LIST:TRIGger:SOURce BUS:TRIGger[:SEQUence]:SOURce IMMEDIATE:INITiate:CONTinuous[:ALL] ON  [:SOURCE]:SWEep:POINts &lt;val&gt;</code>
*☆	Power step size	Y	Y	<code>[[:SOURCE]:POWer[:LEVel][:IMMEDIATE] [:AMPLitude]:STEP[:INCRement] &lt;val&gt;</code>
ST	Sweep time	Y	Y	<code>[[:SOURCE]:SWEep:MODE AUTO [:SOURCE]:SWEep:TIME &lt;val&gt; &lt;unit&gt;</code>
SV	Save state	Y	Y	<code>*SAV &lt;reg_num&gt;[, &lt;seq_num&gt;]</code>

**Table 7-1 8340B/41B Prog. Codes & Equivalent SCPI Sequences (Continued)**

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SW0	Swap network analyzer channels	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SW1	Swap network analyzer channels	Y	Y	<i>supported, but no equivalent SCPI command sequence</i>
SX	External sweep type	N	Y	<i>supported, but has no effect on PSG</i>
T1	Free run sweep trigger mode	Y	Y	:TRIGger[:SEquence]:SOURce IMMEDIATE :INITiate:CONTinuous[:ALL] ON
T2	Line sweep trigger mode	N	N	<i>not supported</i>
T3	External sweep trigger mode	Y	Y	:TRIGger[:SEquence]:SOURce EXTERNAL :INITiate:CONTinuous[:ALL] ON
T4	Single sweep trigger mode	N	Y	:INITiate[:IMMEDIATE][:ALL]
TL	Sweep time limit	Y	Y	[:SOURce]:SWEep:TIME:LLIMit <val> <unit>
TS	Take sweep	Y	Y	:TSWEEP
UP	Step up (increments active function by step value)	Y	Y	<i>supported, see Table 6-2 on page 234</i>
VR	CW vernier	N	Y	<i>supported, but has no effect on PSG</i>



Table 7-2 8340 and 8757 Code Compatibility

Code	Sets Active Function	Comp. with OA/OP	Comp. with UP/DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA/OP query and UP/DN command
A2	✓	✓	✓		[ :SOURCE ] : POWER : ALC : SOURCE : EXTERNAL : COUPLING ? [ :SOURCE ] : POWER : ATTENUATION UP [ :SOURCE ] : POWER : ATTENUATION DOWN
AT	✓	✓	✓		[ :SOURCE ] : POWER : ATTENUATION ? [ :SOURCE ] : POWER : ATTENUATION UP [ :SOURCE ] : POWER : ATTENUATION DOWN
CF	✓	✓			[ :SOURCE ] : FREQUENCY : CENTER ?
CW	✓	✓	✓	✓	[ :SOURCE ] : FREQUENCY [ : CW ] ? [ :SOURCE ] : FREQUENCY [ : CW ] UP [ :SOURCE ] : FREQUENCY [ : CW ] DOWN
DF	✓	✓			[ :SOURCE ] : FREQUENCY : SPAN ?
FA	✓	✓			[ :SOURCE ] : FREQUENCY : START ?
FB	✓	✓			[ :SOURCE ] : FREQUENCY : STOP ?
FM1	✓	✓			[ :SOURCE ] : FM2 [ : DEVIATION ] ?
MA	✓	✓			[ :SOURCE ] : MARKER0 : FREQUENCY ?
M1	✓	✓			[ :SOURCE ] : MARKER1 : FREQUENCY ?
M2	✓	✓			[ :SOURCE ] : MARKER2 : FREQUENCY ?
M3	✓	✓			[ :SOURCE ] : MARKER3 : FREQUENCY ?
M4	✓	✓			[ :SOURCE ] : MARKER4 : FREQUENCY ?
M5	✓	✓			[ :SOURCE ] : MARKER5 : FREQUENCY ?
M6	✓	✓			[ :SOURCE ] : MARKER6 : FREQUENCY ?
M7	✓	✓			[ :SOURCE ] : MARKER7 : FREQUENCY ?
M8	✓	✓			[ :SOURCE ] : MARKER8 : FREQUENCY ?
M9	✓	✓			[ :SOURCE ] : MARKER9 : FREQUENCY ?
PL	✓	✓	✓	✓	[ :SOURCE ] : POWER [ : LEVEL ] [ : IMMEDIATE ] [ : AMPLITUDE ] ? [ :SOURCE ] : POWER [ : LEVEL ] [ : IMMEDIATE ] [ : AMPLITUDE ] UP [ :SOURCE ] : POWER [ : LEVEL ] [ : IMMEDIATE ] [ : AMPLITUDE ] DOWN
PS	✓	✓			[ :SOURCE ] : POWER : SPAN ?
RC	✓				<i>none</i>

Table 7-2 8340 and 8757 Code Compatibility (Continued)

Code	Sets Active Function	Comp. with OA/OP	Comp. with UP/DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA/OP query and UP/DN command
SB	✓	✓			<i>supported, but no equivalent SCPI command sequence</i>
SF	✓	✓		✓	[:SOURCE]:FREQUENCY[:CW]:STEP[:INCREMENT]?
SHA1	✓	✓	✓	✓	[:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE] UP [:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE] DOWN
SHA2	✓	✓		✓	[:SOURCE]:POWER:ALC:LEVEL?
SHA3	✓	✓	✓	✓	[:SOURCE]:POWER:ALC:LEVEL? [:SOURCE]:POWER:ATTENUATION UP [:SOURCE]:POWER:ATTENUATION DOWN
SHAZ	✓	✓		✓	[:SOURCE]:POWER:ALC:LEVEL?
SHCF	✓	✓		✓	[:SOURCE]:FREQUENCY[:CW]:STEP[:INCREMENT]?
SHCW	✓	✓			[:SOURCE]:FREQUENCY:START? or [:SOURCE]:FREQUENCY:STOP?
SHFA	✓	✓		✓	[:SOURCE]:FREQUENCY:MULTIPLIER?
SHFB	✓	✓		✓	[:SOURCE]:FREQUENCY:OFFSET?
SHPL	✓	✓	✓	✓	[:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE]:STEP[:INCREMENT]? [:SOURCE]:POWER:ATTENUATION UP [:SOURCE]:POWER:ATTENUATION DOWN
SHPS	✓	✓	✓	✓	[:SOURCE]:POWER:ALC:LEVEL? [:SOURCE]:POWER:ATTENUATION UP [:SOURCE]:POWER:ATTENUATION DOWN
SHRF	✓	✓	✓	✓	[:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE] UP [:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE] DOWN
SHSL	✓	✓			[:SOURCE]:POWER:ATTENUATION?
SHSN	✓	✓		✓	[:SOURCE]:SWEPT:POINTS?
SL	✓	✓			[:SOURCE]:POWER:SLOPE?
SM	✓	✓			[:SOURCE]:FREQUENCY:MANUAL?
SN	✓	✓		✓	[:SOURCE]:SWEPT:POINTS?
SP	✓	✓		✓	[:SOURCE]:POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE]:STEP[:INCREMENT]?

**Table 7-2 8340 and 8757 Code Compatibility (Continued)**

Code	Sets Active Function	Comp. with OA/OP	Comp. with UP/DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA/OP query and UP/DN command
ST	✓	✓			[ :SOURCE ] :SWEep:TIME?
SV	✓				<i>none</i>
TL	✓	✓			[ :SOURCE ] :SWEep:TIME:LLIMit?

## 836xxB/L Compatible SCPI Commands

Table 7-3 is a comprehensive list of 836xxB/L SCPI commands arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

The preset state of the PSG differs from that of the 836xxB/L. The RF output and sweep are turned off in the PSG, while in the 836xxB/L, these parameters are turned on. To optimize the benefit of using 836xxB/L compatible commands with a PSG, set up a user-defined preset state, emulating the preset state of the 836xxB/L.

To use the commands, select **8360** as the remote language. See “:LANGuage” on page 93 for selecting the language type.

When using the programming codes in this section, you can:

- set the PSG system language to 8360 Series for the current session:

**Utility > GPIB/RS-232 LAN > Remote Language > 8360 Series**

or send the command:

```
:SYST:LANG "8360"
```

- set the PSG system language to 8360 so that it does not reset with either preset, instrument power cycle or \*RST command:

**Utility > Power On/Preset > Preset Language > 8360 Series**

or send the command:

```
:SYST:PRESET:LANG "8360"
```

- set the \*IDN? response to any 8360-like response you prefer. Refer to the :SYSTEM:IDN command on page 347.

---

**NOTE** Some of the PSG supported commands are a subset of the 836xxB/L commands. When this occurs, the syntax supported by the PSG is shown in addition to the syntax that is not supported

---

**Table 7-3 836xxB/L SCPI Commands**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
<b>IEEE Common Commands</b>		
*CLS	Y	Y
*ESE <data>	Y	Y
*ESE?	Y	Y
*ESR?	Y	Y
*IDN? <sup>a</sup>	Y	Y
*LRN?	N	N
*OPC	Y	Y
*OPC?	Y	Y
*OPT?	N	N
*RCL <reg_num>	Y	Y
*RST	Y	Y
*SAV <reg_num>	Y	Y
*SRE <data>	Y	Y
*SRE?	Y	Y
*STB?	Y	Y
*TRG	Y	Y
*TST?	Y	Y
*WAI	Y	Y
<b>Abort Subsystem</b>		
:ABORT	Y	Y
<b>Amplitude Modulation Subsystem</b>		
:AM[:DEPTh] <num>[PCT] MAXimum MINimum <num>DB	Y	
:AM[:DEPTh]? [MAXimum MINimum]	Y	

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:AM:INTernal:FREQUency <num>[<freq suffix>] MAXimum MINimum	Y	
:AM:INTernal:FREQUency? [MAXimum MINimum]	Y	
:AM:INTernal:FUNcTion SINusoid SQUare TRIangle RAMP NOISE	Y	
:AM:INTernal:FUNcTion?	Y	
:AM:SOURce INTernal EXTernal	Y	
:AM:SOURce?	Y	
:AM:MODE DEEP NORMAL	Y	
:AM:MODE?	Y	
:AM:STATE ON OFF 1 0	Y	
:AM:STATE?	Y	
:AM:TYPE LINear EXPonential	Y	
:AM:TYPE?	Y	
<b>Calibration Subsystem</b>		
:CALibration:AM:AUTO ON OFF 1 0	N	
:CALibration:AM:AUTO?	N	
:CALibration:AM[:EXECute]	N	
:CALibration:PEAKing:AUTO ON OFF 1 0	N	N
:CALibration:PEAKing:AUTO?	N	N
:CALibration:PEAKing[:EXECute]	N	N
:CALibration:PMETer:DETEctor:INITiate? IDETEctor DIODE	N	N
:CALibration:PMETer:DETEctor:NEXT? <num>[<lvl suffix>]	N	N
:CALibration:PMETer:FLATness:INITiate? USER DIODE PMETER MMHead	N	N
:CALibration:PMETer:FLATness:NEXT? <value>[<lvl suffix>]	N	N
:CALibration:SPAN:AUTO ON OFF 1 0	N	N

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:CALibration:SPAN:AUTO?	N	N
:CALibration:SPAN[:EXECute]	N	N
:CALibration:TRACk	N	N
<b><i>Correction Subsystem</i></b>		
:CORRection:ARRAy[i]{<value>[DB]}	N	N
:CORRection:ARRAy[i]?	N	N
:CORRection:FLATness {<num>[freq suffix],<num>[DB]}2*801	N	N
:CORRection:FLATness?	Y	Y
:CORRection:SOURce[i] ARRAy FLATness	N	N
:CORRection:SOURce[i]?	N	N
:CORRection:FLATness:POINts? [MAXimum MINimum]	Y	Y
:CORRection[:STATe] ON OFF 1 0	Y	Y
:CORRection[:STATe]?	Y	Y
<b><i>Diagnostics Subsystem</i></b>		
:DIAGnostics:ABUS? <value>	N	N
:DIAGnostics:ABUS:AVERAge <value>	N	N
:DIAGnostics:ABUS:AVERAge?	N	N
:DIAGnostics:ABUS:STATus?	N	N
:DIAGnostics:INSTRument:PMETer:ADDRess <value>	N	N
:DIAGnostics:INSTRument:PMETer:ADDRess?	N	N
:DIAGnostics:INSTRument:PRINter:ADDRess <value>	N	N
:DIAGnostics:INSTRument:PRINter:ADDRess?	N	N
:DIAGnostics:IORW <value>,<value>	N	N
:DIAGnostics:IORW? <value>	N	N
:DIAGnostics:OUTPut:FAULt?	N	N

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:DIAGnostics:RESult?	N	N
:DIAGnostics:TEST:CONTinue	N	N
:DIAGnostics:TEST:DATA:DESC?	N	N
:DIAGnostics:TEST:DATA:MAXimum?	N	N
:DIAGnostics:TEST:DATA:MINimum?	N	N
:DIAGnostics:TEST:DATA:VALue?	N	N
:DIAGnostics:TEST:DISable {<num>}1*? ALL	N	N
:DIAGnostics:TEST:ENABle {<num>}1*? ALL	N	N
:DIAGnostics:TEST[:EXECute] <value>	N	N
:DIAGnostics:TEST:LOG:SOURce ALL FAIL	N	N
:DIAGnostics:TEST:LOG:SOURce?	N	N
:DIAGnostics:TEST:LOG[:STATe]?	N	N
:DIAGnostics:TEST:LOG[:STATe] ON OFF 1 0	N	N
:DIAGnostics:TEST:LOOP ON OFF 1 0	N	N
:DIAGnostics:TEST:LOOP?	N	N
:DIAGnostics:TEST:NAME? [<value>]	N	N
:DIAGnostics:TEST:POINTs?	N	N
:DIAGnostics:TEST:RESult? [<value>]	N	N
:DIAGnostics:TINT? <value>	N	N
<b><i>Display Subsystem</i></b>		
:DISPlay[:STATe] ON OFF 1 0	Y	Y
:DISPlay[:STATe]?	Y	Y
<b><i>Frequency Modulation Subsystem</i></b>		
:FM:COUPling AC DC	Y	
:FM:COUPling?	Y	

Table 7-3 836xxB/L SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:FM[:DEVIation] <val><unit> MAXimum MINimum	Y	
:FM[:DEVIation]? [MAXimum MINimum]	Y	
:FM:FILTer:HPASs <num>[<freq suffix>] MAXimum MINimum	N	
:FM:FILTer:HPASs? [MAXimum MINimum]	N	
:FM:INTernal:FREQuency <num>[<freq suffix>] MAXimum  MINimum	Y	
:FM:INTernal:FREQuency? [MAXimum MINimum]	Y	
:FM:INTernal:FUNctIon SINusoid SQUare TRIangle RAMP NOISE	Y	
:FM:INTernal:FUNctIon?	Y	
:FM:SOURce INTernal EXTernal	Y	
:FM:SOURce?	Y	
:FM:SENSitivity <val><freq suffix/V> MAXimum MINimum	Y	
:FM:SENSitivity? [MAXimum MINimum]	Y	
:FM:STATE ON OFF 1 0	Y	
:FM:STATE?	Y	
<b>Frequency Subsystem</b>		
:FREQuency:CENter <num>[<freq suffix>] MAXimum MINimum  UP DOWN	Y	Y
:FREQuency:CENter? [MAXimum MINimum]	Y	Y
:FREQuency[:CW]:FIXed <num>[<freq suffix>] MAXimum  MINimum UP DOWN	Y	Y
:FREQuency[:CW]? [MAXimum MINimum]	Y	Y
:FREQuency[:FIXed]? [MAXimum MINimum]	Y	Y
:FREQuency[:CW]:AUTO ON OFF 1 0	N	N
:FREQuency[:CW]:AUTO?	N	N
:FREQuency[:FIXed]:AUTO ON OFF 1 0	N	N
:FREQuency[:FIXed]:AUTO?	N	N



**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:FREQuency:MANual <num>[<freq suffix>] MAXimum MINimum UP DOWN	N	N
:FREQuency:MANual? [MAXimum MINimum]	N	N
:FREQuency:MODE FIXEd CW SWEp LIST	Y	Y
:FREQuency:MODE?	Y	Y
:FREQuency:MULTiplier <num> MAXimum MINimum <sup>b</sup>	Y	Y
:FREQuency:MULTiplier? [MAXimum MINimum]	Y	Y
:FREQuency:MULTiplier:STATe ON OFF 1 0	N	N
:FREQuency:MULTiplier:STATe?	N	N
:FREQuency:OFFSet <num> MAXimum MINimum	Y	Y
:FREQuency:OFFSet? [MAXimum MINimum]	Y	Y
:FREQuency:OFFSet:STATe ON OFF 1 0	Y	Y
:FREQuency:OFFSet:STATe?	Y	Y
:FREQuency:SPAN <num>[<freq suffix>] MAXimum MINimum UP DOWN	Y	Y
:FREQuency:SPAN? [MAXimum MINimum]	Y	Y
:FREQuency:START <num>[<freq suffix>] MAXimum MINimum UP DOWN	Y	Y
:FREQuency:START? [MAXimum MINimum]	Y	Y
:FREQuency:STEP:AUTO ON OFF 1 0	Y	Y
:FREQuency:STEP:AUTO?	Y	Y
:FREQuency:STEP[: INCRement] <num>[<freq suffix>] MAXimum MINimum	Y	Y
:FREQuency:STEP[: INCRement]?	Y	Y
:FREQuency:STOP <num>[<freq suffix>] MAXimum MINimum UP DOWN	Y	Y
:FREQuency:STOP? [MAXimum MINimum]	Y	Y

Table 7-3 836xxB/L SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
<b>Initiate Subsystem</b>		
:INITiate:CONTinuous ON OFF 1 0	Y	Y
:INITiate:CONTinuous?	Y	Y
:INITiate[:IMMediate]	Y	Y
<b>List Subsystem</b>		
:LIST:DWELl {<num>[<time suffix>] MAXimum MINimum}	Y	Y
:LIST:DWELl? [MAXimum MINimum]	Y	Y
:LIST:DWELl:POINts? [MAXimum MINimum]	Y	Y
:LIST:FREQuency {<value>[<freq suffix>] MAXimum MINimum}	Y	Y
:LIST:FREQuency?	Y	Y
:LIST:FREQuency:POINts? [MAXimum MINimum]	Y	Y
:LIST:MANual <num>	Y	Y
:LIST:MANual?	Y	Y
:LIST:MODE AUTO MANual	Y	Y
:LIST:MODE?	Y	Y
:LIST[:POWer]:CORRection {<value>[DB] MAXimum MINimum}	N	N
:LIST[:POWer]:CORRection?	N	N
:LIST[:POWer]:CORRection:POINts? [MAXimum MINimum]	N	N
:LIST:TRIGger:SOURce IMMEDIATE BUS EXTernal	Y	Y
:LIST:TRIGger:SOURce?	Y	Y
<b>Marker Subsystem</b>		
:MARKer[n]:AMPLitude[:STATe] ON OFF 1 0	N	N
:MARKer[n]:AMPLitude[:STATe]?	N	N
:MARKer[n]:AMPLitude:VALue <value>[DB] MAXimum MINimum	N	N
:MARKer[n]:AMPLitude:VALue? [MAXimum MINimum]	N	N

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:MARKer[n]:AOFF	N	N
:MARKer[n]:DELTA? <value>, <value>	N	N
:MARKer[n]:FREQuency <value>[<freq suffix>] MAXimum MINimum	N	N
:MARKer[n]:FREQuency? [MAXimum MINimum]	N	N
:MARKer[n]:MODE FREQuency DELTA	N	N
:MARKer[n]:MODE?	N	N
:MARKer[n]:REFErence <n>	N	N
:MARKer[n]:REFErence?	N	N
:MARKer[n][:STATE] ON OFF 1 0	N	N
:MARKer[n][:STATE]?	N	N
<b>Measure Subsystem</b>		
:MEASure:AM?	N	
:MEASure:FM?	N	
<b>Modulation Subsystem</b>		
:MODulation:OUTPut:SOURce AM FM	N	
:MODulation:OUTPut:SOURce?	N	
:MODulation:OUTPut:STATE ON OFF 1 0	Y	
:MODulation:OUTPut:STATE?	Y	
:MODulation:STATE?	Y	
<b>Power Subsystem</b>		
:POWer:ALC:BANDwidth :BWIDth <value>[<freq suffix>] MAXimum MINimum	Y	Y
:POWer:ALC:BANDwidth? :BWIDth? [MAXimum MINimum]	Y	Y
:POWer:ALC:BANDwidth :BWIDth:AUTO ON OFF 1 0	Y	Y
:POWer:ALC:BANDwidth :BWIDth:AUTO?	Y	Y

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:POWER:ALC:CFACTOR <value>[DB] MAXimum MINimum UP DOWN	Y	Y
:POWER:ALC:CFACTOR? [MINimum MAXimum]	Y	Y
:POWER:ALC:SOURce PMETer	N	N
:POWER:ALC:SOURce INTernal DIODE MMHead	Y	Y
:POWER:ALC:SOURce?	Y	Y
:POWER:ALC[:STATE] ON OFF 1 0	Y	Y
:POWER:ALC[:STATE]?	Y	Y
:POWER:AMPLifier:STATE ON OFF 1 0	N	N
:POWER:AMPLifier:STATE?	N	N
:POWER:AMPLifier:STATE:AUTO ON OFF 1 0	N	N
:POWER:AMPLifier:STATE:AUTO?	N	N
:POWER:ATTenuation <num>[DB] MAXimum MINimum UP DOWN	Y	Y
:POWER:ATTenuation? [MAXimum MINimum]	Y	Y
:POWER:ATTenuation:AUTO ON OFF 1 0	Y	Y
:POWER:ATTenuation:AUTO?	Y	Y
:POWER:CENTER <num>[<lvl suffix>] MAXimum MINimum UP DOWN	Y	Y
:POWER:CENTER? [MAXimum MINimum]	Y	Y
:POWER[:LEVEL] <num>[<lvl suffix>] MAXimum MINimum UP DOWN	Y	Y
:POWER[:LEVEL]? [MAXimum MINimum]	Y	Y
:POWER:MODE FIXed SWEep	Y	Y
:POWER:MODE?	Y	Y
:POWER:OFFSet <num>[DB] MAXimum MINimum UP DOWN	Y	Y
:POWER:OFFSet? [MAXimum MINimum]	Y	Y
:POWER:OFFSet:STATE ON 1 <sup>c</sup>	N	N
:POWER:OFFSet:STATE OFF 0 <sup>d</sup>	Y	Y

**Table 7-3 836xxB/L SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83620B &amp; 83640B</b>	<b>83620L &amp; 83640L</b>
:POWER:OFFSet:STATe?	Y	Y
:POWER:RANGe <value>[<lvl suffix>] MAXimum MINimum UP DOWN	N	N
:POWER:RANGe?	N	N
:POWER:SEARCh ON OFF 1 0 ONCE	Y	Y
:POWER:SEARCh?	Y	Y
:POWER:SLOPe <value>[DB/<freq suffix>] MIN MAX UP DOWN	Y	Y
:POWER:SLOPe? [MAXimum MINimum]	Y	Y
:POWER:SLOPe:STATe ON OFF 1 0	Y	Y
:POWER:SLOPe:STATe?	Y	Y
:POWER:SPAN <value>[DB] MAXimum MINimum UP DOWN	Y	Y
:POWER:SPAN? [MAXimum MINimum]	Y	Y
:POWER:STARt <val><unit> MAXimum MINimum UP DOWN	Y	Y
:POWER:STARt? [MAXimum MINimum]	Y	Y
:POWER:STATe ON OFF 1 0	Y	Y
:POWER:STATe?	Y	Y
:POWER:STEP:AUTO ON OFF 1 0	Y	Y
:POWER:STEP:AUTO?	Y	Y
:POWER:STEP[:INCRement] <num>[DB] MAXimum MINimum	Y	Y
:POWER:STEP[:INCRement]? [MAXimum MINimum]	Y	Y
:POWER:STOP <val><unit> MAXimum MINimum UP DOWN	Y	Y
:POWER:STOP? [MAXimum MINimum]	Y	Y
<b><i>Pulse Modulation Subsystem</i></b>		
:PULM:EXTernal:DELay <value>[<time suffix>] MAXimum MINimum	N	
:PULM:EXTernal:DELay? [MAXimum MINimum]	N	

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:PULM:EXTErnal:POLarity NORMal INVerted	Y	
:PULM:EXTErnal:POLarity?	Y	
:PULM:INTernAl:FREQuency <num>[<freq suffix>] MAXimum  MINimum	Y	
:PULM:INTernAl:FREQuency? [MAXimum MINimum]	Y	
:PULM:INTernAl:GATE ON OFF 1 0	N	
:PULM:INTernAl:GATE?	N	
:PULM:INTernAl:PERiod <num>[<time suffix>] MAXimum  MINimum	Y	
:PULM:INTernAl:PERiod? [MAXimum MINimum]	Y	
:PULM:INTernAl:TRIGger:SOURce INTernAl EXTernAl	Y	
:PULM:INTernAl:TRIGger:SOURce? [INTernAl EXTernAl]	Y	
:PULM:INTernAl:WIDTh <num>[<time suffix>] MAXimum MINimum	Y	
:PULM:INTernAl:WIDTh? [MAXimum MINimum]	Y	
:PULM:SLEW <value>[<time suffix>] MAXimum MINimum	N	
:PULM:SLEW? [MAXimum MINimum]	N	
:PULM:SLEW:AUTO ON OFF 1 0	N	
:PULM:SLEW:AUTO?	N	
:PULM:SOURce SCALar	N	
:PULM:SOURce INTernAl EXTernAl	Y	
:PULM:SOURce?	Y	
:PULM:STATe ON OFF 1 0	Y	
:PULM:STATe?	Y	
<b>Pulse Subsystem</b>		
:PULSe:FREQuency <num>[<freq suffix>] MAXimum MINimum	Y	
:PULSe:FREQuency? [MAXimum MINimum]	Y	
:PULSe:PERiod <num>[<time suffix>] MAXimum MINimum	Y	

**Table 7-3 836xxB/L SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83620B &amp; 83640B</b>	<b>83620L &amp; 83640L</b>
:PULSe:PERiod? [MAXimum MINimum]	Y	
:PULSe:WIDTh <num>[<time suffix>] MAXimum MINimum	Y	
:PULSe:WIDTh? [MAXimum MINimum]	Y	
<b><i>Reference Oscillator Subsystem</i></b>		
:ROSCillator:SOURce?	Y	Y
:ROSCillator:SOURce:AUTO ON OFF 1 0	Y	Y
:ROSCillator:SOURce:AUTO?	Y	Y
:ROSCillator:SOURce INTernal EXTernal NONE	Y	Y
<b><i>Status Subsystem</i></b>		
:STATus:OPERation:CONDition?	Y	Y
:STATus:OPERation:ENABle <value>	Y	Y
:STATus:OPERation:ENABle?	Y	Y
:STATus:OPERation[:EVENT]?	Y	Y
:STATus:OPERation:NTRansition <value>	Y	Y
:STATus:OPERation:NTRansition?	Y	Y
:STATus:OPERation:PTRansition <value>	Y	Y
:STATus:OPERation:PTRansition?	Y	Y
:STATus:PRESet	Y	Y
:STATus:QUESTionable:CONDition?	Y	Y
:STATus:QUESTionable:ENABle <value>	Y	Y
:STATus:QUESTionable:ENABle?	Y	Y
:STATus:QUESTionable[:EVENT]?	Y	Y
:STATus:QUESTionable:NTRansition <value>	Y	Y
:STATus:QUESTionable:NTRansition?	Y	Y
:STATus:QUESTionable:PTRansition <value>	Y	Y

Table 7-3 836xxB/L SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:STATus:QUESTionable:PTRansition?	Y	Y
<b><i>Sweep Subsystem</i></b>		
:SWEep:CONTrol:STATe ON OFF 1 0	N	N
:SWEep:CONTrol:STATe?	N	N
:SWEep:CONTrol:TYPE MASTER SLAVE	N	N
:SWEep:CONTrol:TYPE?	N	N
:SWEep:DWELL <num>[<time suffix>] MAXimum MINimum	Y	Y
:SWEep:DWELL? [MAXimum MINimum]	Y	Y
:SWEep:DWELL:AUTO ON OFF 1 0	N	N
:SWEep:DWELL:AUTO?	N	N
:SWEep:GENeration STEPPed ANALog	N	N
:SWEep:GENeration?	N	N
:SWEep:MANual:POINT <num> MAXimum MINimum	Y	Y
:SWEep:MANual:POINT? [MAXimum MINimum]	Y	Y
:SWEep:MANual[:RELative] <value>	N	N
:SWEep:MANual[:RELative]?	N	N
:SWEep:MARKer:STATe ON OFF 1 0	N	N
:SWEep:MARKer:STATe?	N	N
:SWEep:MARKer:XFER	N	N
:SWEep:MODE AUTO MANual	Y	Y
:SWEep:MODE?	Y	Y
:SWEep:POINTs <num> MAXimum MINimum	Y	Y
:SWEep:POINTs? [MAXimum MINimum]	Y	Y
:SWEep:STEP <value>[<freq suffix>] MAXimum MINimum	N	N
:SWEep:STEP? [MAXimum MINimum]	N	N



**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:SWEep:TIME <value>[<time suffix>] MAXimum MINimum	N	N
:SWEep:TIME? [MAXimum MINimum]	N	N
:SWEep:TIME:AUTO ON OFF 1 0	N	N
:SWEep:TIME:AUTO?	N	N
:SWEep:TIME:LLIMit <value>[<time suffix>] MAXimum MINimum	N	N
:SWEep:TIME:LLIMit? [MAXimum MINimum]	N	N
:SWEep:TRIGger:SOURce IMMEDIATE BUS EXTERNAL	Y	Y
:SWEep:TRIGger:SOURce?	Y	Y
<b>System Subsystem</b>		
:SYSTem:ALTErnate <value> MAXimum MINimum	N	N
:SYSTem:ALTErnate? [MAXimum MINimum]	N	N
:SYSTem:ALTErnate:STATe ON OFF 1 0	N	N
:SYSTem:ALTErnate:STATe?	N	N
:SYSTem:COMMunicate:GPIB:ADDRESS <number>	Y	Y
:SYSTem:DUMP:PRINter?	N	N
:SYSTem:ERRor?	Y	Y
:SYSTem:LANGuage CIIL COMPAtible	N	N
:SYSTem:LANGuage SCPI	Y	Y
:SYSTem:MMHead:SELEct:AUTO ON OFF 1 0	Y	Y
:SYSTem:MMHead:SELEct:AUTO?	Y	Y
:SYSTem:MMHead:SELEct FRONT REAR NONE <sup>e</sup>	Y	Y
:SYSTem:MMHead:SELEct?	Y	Y
:SYSTem:PRESet[:EXECute]	Y	Y
:SYSTem:PRESet:SAVE	Y	Y
:SYSTem:PRESet:TYPE FACTory USER	Y	Y
:SYSTem:PRESet:TYPE?	Y	Y

**Table 7-3 836xxB/L SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:SYSTem:SECurity:COUNT <value> <sup>fg</sup>	Y	Y
:SYSTem:SECurity:COUNT? [MINimum MAXimum]	Y	Y
:SYSTem:SECurity[:STATe] ON OFF 1 0 <sup>e</sup>	Y	Y
:SYSTem:SECurity[:STATe]?	Y	Y
:SYSTem:VERSion?	Y	Y
<b><i>Trigger Subsystem</i></b>		
:TRIGger[:IMMediate]	Y	Y
:TRIGger:ODELay <value>[time suffix] MAXimum MINimum	N	N
:TRIGger:ODELay? [MAXimum MINimum]	N	N
:TRIGger:SOURce IMMEDIATE BUS EXTernal	Y	Y
:TRIGger:SOURce?	Y	Y
<b><i>Tsweep Subsystem</i></b>		
:TSWEEP	N	N
<b><i>Unit Subsystem</i></b>		
:UNIT:AM DB PCT	N	
:UNIT:AM?	N	
:UNIT:POWer {<lvl suffix>}	Y	Y
:UNIT:POWer?	Y	Y

- a. The identification information can be modified for the PSG to reflect the signal generator that is being replaced. Refer to “:SYSTem:IDN” on page 347 for more information.
- b. A multiplier of zero is not allowed.
- c. The PSG will accept this command, but it has no effect.
- d. This command resets the power offset level to 0dBm. It does not turn off or disable the power offset feature.
- e. Since the PSG does not have a front panel millimeter head (source module) interface connector, the “FRONT” suffix defaults to the rear connector.
- f. Flash memory allows only a limited number of “writes and erasures”, excessive use of this command will reduce the memory lifetime.
- g. This command can take several hours to execute because the PSG memory size is much larger than the HP 836xx memory.

## 8373xB and 8371xB Compatible SCPI Commands

Table 7-4 is a comprehensive list of 8373xB and 8371xB SCPI commands arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

To use the commands, select **8371xB or 8373xB** as the remote language. See “:LANGuage” on page 93 for selecting the language type.

When using the programming codes in this section, you can:

- set the PSG system language to 8371xB or 8373xB for the current session:

**Utility > GPIB/RS-232 LAN > Remote Language > 8371xB or 8373xB**

or send the command:

:SYST:LANG "83712" or "83732"

- set the PSG system language to 8360 so that it does not reset with either preset, instrument power cycle or \*RST command:

**Utility > Power On/Preset > Preset Language > 8360 Series**

or send the command:

:SYST:PRESET:LANG "83712" or "83732"

- set the \*IDN? response to any 8373xB- or 8371xB-like response you prefer. Refer to the :SYSTem:IDN command on page 347.

---

**NOTE** Some of the PSG supported commands are subsets of the 8373xB and 8371xB commands. When this occurs, the syntax supported by the PSG is shown in addition to the syntax that is not supported.

---

**Table 7-4 8373xB and 8371xB SCPI Commands**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
<i>IEEE Common Commands</i>		
*CLS	Y	Y
*DMC	N	N
*EMC	N	N
*EMC?	N	N
*ESE <data>	Y	Y
*ESE?	Y	Y
*ESR?	Y	Y
*GMC?	N	N

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
*IDN? <sup>a</sup>	Y	Y
*LMC?	N	N
*LRN?	N	N
*OPC	Y	Y
*OPC?	Y	Y
*OPT?	N	N
*PMC	N	N
*PSC	Y	Y
*PSC?	Y	Y
*RCL <reg_num>	Y	Y
*RMC	N	N
*RST	Y	Y
*SAV <reg_num>	Y	Y
*SRE <data>	Y	Y
*SRE?	Y	Y
*STB?	Y	Y
*TST?	Y	Y
*WAI	Y	Y
<b><i>Abort Subsystem</i></b>		
:ABORt	Y	
<b><i>Amplitude Modulation Subsystem</i></b>		
[ :SOURce]:AM[:DEPTh] <val><unit> <sup>b</sup>	Y	
[ :SOURce]:AM[:DEPTh] <num>[<PCT>]   <num>DB	Y	
[ :SOURce]:AM[:DEPTh]:STEP[:INCRement] incr MINimum MAXimum DEFault	Y	
[ :SOURce]:AM:INTernal:FREQUency <num>[<freq suffix>] incr MINimum MAXimum DEFault	Y	
[ :SOURce]:AM:INTernal:FREQUency:STEP[:INCRement]	Y	

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83731B &amp; 83732B</b>	<b>83711B &amp; 83712B</b>
[ :SOURce]:AM:INTernal:FUNctIon SINusoid SQUare TRIangle  RAMP NOISe UNIForm GAUSSian	Y	
[ :SOURce]:AM:SENSitivity <val> MIN MAX DEF	N	
[ :SOURce]:AM:SOURce FEED [ :SOURce]:AM:SOURce INTernal EXTernal	N Y	
[ :SOURce]:AM:SOURce?	Y	
[ :SOURce]:AM:STATe ON OFF	Y	
[ :SOURce]:AM:STATe?	Y	
[ :SOURce]:AM:TYPE LINear EXPonential	Y	
[ :SOURce]:AM:TYPE?	Y	
<b>Display Subsystem</b>		
:DISPlay[:WINDow][:STATe] ON OFF 1 0	Y	Y
:DISPlay[:WINDow][:STATe]?	Y	Y
<b>Initiate Subsystem</b>		
:INITiate:CONTInuous ON OFF 1 0	Y	
:INITiate:CONTInuous?	Y	
<b>Correction Subsystem</b>		
[ :SOURce]:CORRection:FLATness[:DATA] <freq>,<corr.>,... <freq>,<corr.>	Y	Y
[ :SOURce]:CORRection:FLATness:POINts <points>	Y	Y
[ :SOURce]:CORRection[:STATe] ON OFF	Y	Y
[ :SOURce]:CORRection[:STATe]?	Y	Y
[ :SOURce]:CORRection:CSET[:SElect] tableno	N	N
[ :SOURce]:CORRection:CSET[:SElect]?	N	N
[ :SOURce]:CORRection:CSET:STATe ON OFF 1 0	N	N
[ :SOURce]:CORRection:CSET:STATe?	N	N
<b>Frequency Modulation Subsystem</b>		
[ :SOURce]:FM:COUpling AC DC	Y	
[ :SOURce]:FM:COUpling?	Y	

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[ :SOURce]:FM[:DEVIation] <val><unit>	Y	
[ :SOURce]:FM[:DEVIation]:STEP[:INCRement] <val> [<freq suffix>]	Y	
[ :SOURce]:FM:INTErnal:FREQuency <num>[<freq suffix>]	Y	
[ :SOURce]:FM:INTErnal:FREQuency:STEP[:INCRement] incr  MINimum MAXimum DEFault	N	
[ :SOURce]:FM:INTErnal:FUNCTion SINusoid SQUAre TRIANGLE  RAMP UNIForm GAUSSian	N	
[ :SOURce]:FM:SENSitivity?	Y	
[ :SOURce]:FM:SOURce FEED [ :SOURce]:FM:SOURce INTErnal EXTErnal	N Y	
[ :SOURce]:FM:STATe ON OFF 1 0	Y	
[ :SOURce]:FM:STATe?	Y	
<b>Frequency Subsystem</b>		
[ :SOURce]:FREQuency[:CW]:FIXed <num>[<freq suffix>] UP  DOWN DEFault	Y	Y
[ :SOURce]:FREQuency[:CW]:FIXed [MAXimum MINimum DEFault]	Y	Y
[ :SOURce]:FREQuency[:CW]:FIXed:STEP <val><unit>	Y	Y
[ :SOURce]:FREQuency[:CW]:FIXed:STEP?	Y	Y
[ :SOURce]:FREQuency:MULTiplier <val> UP DOWN DEFault <sup>c</sup>	Y	Y
[ :SOURce]:FREQuency:MULTiplier?	Y	Y
[ :SOURce]:FREQuency:MULTiplier:STEP[:INCRement] incr  MINimum MAXimum DEFault	N	N
[ :SOURce]:FREQuency:MULTiplier:STEP[:INCRement]?	N	N
<b>Memory Subsystem</b>		
:MEMory:CATalog[:ALL]?	Y	Y
:MEMory:CATalog:TABLE?	N	N
:MEMory:CATalog:MACRo	N	N
:MEMory:RAM:INITialize	N	N
:MEMory:TABLE:FREQuency freq,...freq MINimum MAXimum	N	N

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
:MEMory:TABLE:FREQuency? MINimum MAXimum	N	N
:MEMory:TABLE:FREQuency:POINts?	N	N
:MEMory:TABLE:LOSS[:MAGNitude] cf,...cf MINimum MAXimum	N	N
:MEMory:TABLE:LOSS[:MAGNitude]?	N	N
:MEMory:TABLE:LOSS[:MAGNitude]:POINts?	N	N
:MEMory:TABLE:SElect tableno	N	N
:MEMory:TABLE:SElect?	N	N
<b>Modulation Subsystem</b>		
[:SOURce]:MODulation:AOFF	Y	
[:SOURce]:MODulation:STATe ON OFF	N	
[:SOURce]:MODulation:STATe?	Y	
<b>Output Subsystem</b>		
:OUTPut:IMPedance?	N	N
:OUTPut:PROTection[:STATe] ON OFF	Y	Y
:OUTPut:PROTection[:STATe]?	Y	Y
:OUTPut[:STATe] ON OFF 1 0	Y	Y
:OUTPut[:STATe]?	Y	Y
<b>Phase Modulation Subsystem</b>		
[:SOURce]:PM:COUpling AC DC	Y	
[:SOURce]:PM[:DEVIation] <val><unit>	Y	
[:SOURce]:PM[:DEVIation]:STEP[:INCRement]	Y	
[:SOURce]:PM:INTernal:FREQuency <val><unit>	Y	
[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement]	Y	
[:SOURce]:PM:INTernal:FUNCTion SINusoid SQUare TRIAnge  RAMP UNIForm GAUSSian	Y	
[:SOURce]:PM:RANGe AUTO LOW HIGH	Y	
[:SOURce]:PM:SENSitivity sens MINimum MAXimum DEFault	N	

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[ :SOURCE ] : PM : SOURCE INTERNAL   FEED   EXTERNAL <sup>d</sup>	Y	
[ :SOURCE ] : PM : STATE ON   OFF   1   0	Y	
<b>Power Subsystem</b>		
[ :SOURCE ] : POWER : ALC : PMETER pmeter   MINimum   MAXimum   DEFAULT	N	N
[ :SOURCE ] : POWER : ALC : PMETER?	N	N
[ :SOURCE ] : POWER : ALC : PMETER : STEP incr   MINimum   MAXimum   DEFAULT	N	N
[ :SOURCE ] : POWER : ALC : PMETER : STEP?	N	N
[ :SOURCE ] : POWER : ALC : SOURCE PMETER [ :SOURCE ] : POWER : ALC : SOURCE INTERNAL   DIODE	N Y	N Y
[ :SOURCE ] : POWER : ALC : SOURCE?	Y	Y
[ :SOURCE ] : POWER : ATTENUATION : AUTO ONCE [ :SOURCE ] : POWER : ATTENUATION : AUTO ON   OFF	N Y	N Y
[ :SOURCE ] : POWER : ATTENUATION : AUTO?	Y	Y
[ :SOURCE ] : POWER [ :LEVEL ] ampl   MINimum   MAXimum   UP   DOWN   DEFAULT	Y	Y
[ :SOURCE ] : POWER [ :LEVEL ]?	Y	Y
[ :SOURCE ] : POWER [ :LEVEL ] : STEP incr   MINimum   MAXimum   DEFAULT	Y	Y
[ :SOURCE ] : POWER [ :LEVEL ] : STEP?	Y	Y
[ :SOURCE ] : POWER : PROTECTION : STATE ON   OFF	Y	Y
[ :SOURCE ] : POWER : PROTECTION : STATE?	Y	Y
<b>Pulse Modulation Subsystem</b>		
[ :SOURCE ] : PULM : EXTERNAL : POLARITY NORMAL   INVERTED	Y	
[ :SOURCE ] : PULM : EXTERNAL : POLARITY?	Y	
[ :SOURCE ] : PULM : SOURCE INTERNAL   EXTERNAL	Y	
[ :SOURCE ] : PULM : SOURCE?	Y	
[ :SOURCE ] : PULM : STATE ON   OFF   1   0	Y	
[ :SOURCE ] : PULM : STATE?	Y	
<b>Pulse Subsystem</b>		



**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83731B &amp; 83732B</b>	<b>83711B &amp; 83712B</b>
[ :SOURce]:PULSe:DElay delay MINimum MAXimum UP DOWN  DEfault	Y	
[ :SOURce]:PULSe:DElay?	Y	
[ :SOURce]:PULSe:DElay:STEP <num>[<time suffix>][DEfault]	Y	
[ :SOURce]:PULSe:DElay:STEP? [DEfault]	Y	
[ :SOURce]:PULSe:DOUble[:STATE] ON OFF	N	
[ :SOURce]:PULSe:DOUble[:STATE]?	N	
[ :SOURce]:PULSe:FREquency freq MINimum MAXimum UP DOWN  DEfault	Y	
[ :SOURce]:PULSe:FREquency?	Y	
[ :SOURce]:PULSe:FREquency:STEP freq DEfault	Y	
[ :SOURce]:PULSe:FREquency:STEP? [MIN MAX DEF]	Y	
[ :SOURce]:PULSe:PERiod <num>[<time suffix>] UP DOWN	Y	
[ :SOURce]:PULSe:PERiod?	Y	
[ :SOURce]:PULSe:PERiod:STEP <num>[<time suffix>]	Y	
[ :SOURce]:PULSe:PERiod:STEP?	Y	
[ :SOURce]:PULSe:TRANSition[:LEADing] SLOW MEDIum FAST	N	
[ :SOURce]:PULSe:TRANSition[:LEADing]?	N	
[ :SOURce]:PULSe:TRANSition:STATe ON OFF	N	
[ :SOURce]:PULSe:TRANSition:STATe?	N	
[ :SOURce]:PULSe:WIDTh MAXimum MINimum UP DOWN DEfault	Y	
[ :SOURce]:PULSe:WIDTh? [MAXimum MINimum DEfault]	Y	
[ :SOURce]:PULSe:WIDTh:STEP <num>[<time suffix>] DEfault	Y	
[ :SOURce]:PULSe:WIDTh:STEP? [MINimum MAXimum DEfault]	Y	
<b>Reference Oscillator Subsystem</b>		
[ :SOURce]:ROScillator:SOURce?	Y	Y
<b>Status Subsystem</b>		
:STATus:OPERation:CONDition?	Y	Y

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
:STATus:OPERation:ENABle <value>	Y	Y
:STATus:OPERation:ENABle?	Y	Y
:STATus:OPERation[:EVENT]?	Y	Y
:STATus:OPERation:NTRansition <value>	Y	Y
:STATus:OPERation:NTRansition?	Y	Y
:STATus:OPERation:PTRansition <value>	Y	Y
:STATus:OPERation:PTRansition?	Y	Y
:STATus:PRESet	Y	Y
:STATus:QUESTionable:CONDition?	Y	Y
:STATus:QUESTionable:ENABle <value>	Y	Y
:STATus:QUESTionable:ENABle?	Y	Y
:STATus:QUESTionable[:EVENT]?	Y	Y
:STATus:QUESTionable:NTRansition <value>	Y	Y
:STATus:QUESTionable:NTRansition?	Y	Y
:STATus:QUESTionable:PTRansition <value>	Y	Y
:STATus:QUESTionable:PTRansition?	Y	Y
<b>System Subsystem</b>		
:SYSTem:COMMunicate:GPIB:ADDRess <number>	Y	Y
:SYSTem:COMMunicate:GPIB:ADDRess?	Y	Y
:SYSTem:COMMunicate:PMETer:ADDRess	Y	Y
:SYSTem:COMMunicate:PMETer:ADDRess?	Y	Y
:SYSTem:ERRor?	Y	Y
:SYSTem:KEY keycode MINimum MAXimum	N	N
:SYSTem:KEY?	N	N
:SYSTem:LANGuage "COMP=8673" "COMPatibility=8673"	N	N
:SYSTem:LANGuage "SCPI"	Y	Y
:SYSTem:LANGuage?	Y	Y
:SYSTem:PRESet	Y	Y

**Table 7-4 8373xB and 8371xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
:SYStem:VERSIon?	Y	Y
<b>Trigger Subsystem</b>		
:TRIGger[:SEquence :START]:SOURce IMMEDIATE EXTERNAL	N	
:TRIGger[:SEquence :START]:SOURCE?	N	
:TRIGger:SEquence2:STOP:SOURce IMMEDIATE EXTERNAL	N	
:TRIGger:SEquence2:STOP:SOURCE?	N	
:TRIGger:SEquence2:SLOPe	N	
<b>Unit Subsystem</b>		
:UNIT:FREQuency {<freq suffix>}	Y	Y
:UNIT:FREQuency?	Y	Y
:UNIT:POWer {<lvl suffix>}	Y	Y
:UNIT:POWer?	Y	Y
:UNIT:TIME	N	N
:UNIT:TIME?	N	N
:UNIT:VOLTage {<lvl suffix>}	N	N
:UNIT:VOLTage?	N	N

a. The identification information can be modified for the PSG to reflect the signal generator that is being replaced. Refer to “:SYStem:IDN” on [page 347](#) for more information.

b. In linear mode, % cannot be used to select percent as the unit. Use PCT to specify percent as the unit.

c. A multiplier of zero is not allowed.

d. If FEED is selected, the query returns INT. FEED and INTERNAL are synonymous.

## 8375xB Compatible SCPI Commands

[Table 7-5](#) is a comprehensive list of 83751B and 83752B SCPI commands, arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

To use the commands, select **8375xB** as the remote language. See “:LANGuage” on [page 93](#) for selecting the language type.

When using the programming codes in this section, you can:

- set the PSG system language to 8375xB for the current session:  
**Utility > GPIB/RS-232 LAN > Remote Language > 8375xB**  
 or send the command:  
 :SYST:LANG "83752"
- set the PSG system language to 8375xB so that it does not reset with either preset, instrument power cycle or \*RST command:  
**Utility > Power On/Preset > Preset Language > 8375xB**  
 or send the command:  
 :SYST:PRESET:LANG "83752"
- set the \*IDN? response to any 8375xB-like response you prefer. Refer to the [:SYSTEM:IDN](#) command on [page 347](#).

---

**NOTE** Some supported commands require the installation of hardware or firmware options.

---

**Table 7-5 8375xB SCPI Commands**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
<i>IEEE Common Commands</i>	
*CLS	Y
*DMC	N
*EMC	N
*EMC?	N
*ESE <value>	Y
*ESE?	Y
*ESR?	Y
*GMC? <label>	N
*IDN?	Y
*LMC?	N
*LRN?	N
*OPC	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
*OPC?	Y
*OPT?	N
*PMC	N
*PSC ON OFF 1 0	Y
*PSC?	Y
*RCL <reg_num>	Y
*RMC <label>	N
*RST	Y
*SAV <reg_num>	Y
*SRE <value>	Y
*SRE?	Y
*STB?	Y
*TRG	Y
*TST?	Y
*WAI	Y
<b><i>Abort Subsystem</i></b>	
:ABORT	Y
<b><i>Amplitude Modulation Subsystem</i></b>	
:AM:SOURce1 INTernal EXTernal	N
:AM:SOURce INTernal EXTernal	Y
:AM:SOURce1?	N
:AM:SOURce?	Y
:AM:STATe ON OFF 1 0	Y
:AM:STATe?	Y
<b><i>Calibration Subsystem</i></b>	
:CALibration:PEAKing[:EXECute]	N

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:CALibration:PEAKing[:EXECute]? <dac_va>	N
:CALibration:PMETer:FLATness:INITiate? USER	N
:CALibration:PMETer:FLATness:NEXT? <value>[<lvlsuffix>]	N
:CALibration:SECurity:CODE <old> <new>	N
:CALibration:SECurity:PASSword <passwd>	N
:CALibration:TRACk	N
<b><i>Correction Subsystem</i></b>	
:CORRection:FLATness:AMPL <value>[DB],<value>[DB]...	N
:CORRection:FLATness:AMPL?	N
:CORRection:FLATness:FREQ <value>[<freqsuffix>],<value>[<freqsuffix>]...	N
:CORRection:FLATness:FREQ?	N
:CORRection:FLATness:POINts? MAXimum MINimum	N
:CORRection:VOLTs:OFFSet	N
:CORRection:VOLTs:OFFSet?	N
:CORRection:VOLTs:SCALe	N
:CORRection:VOLTs:SCALe?	N
:CORRection[:STATe] ON OFF 1 0	Y
:CORRection[:STATe]?	Y
<b><i>Diagnostics Subsystem</i></b>	
:DIAG:LRNS?	N
:DIAGnostic:TEST:FULLtest:REPort?	N
:DIAGnostic:TEST:FULLtest?	N
<b><i>Display Subsystem</i></b>	
:DISPlay[:STATe] ON OFF 1 0	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:DISPlay[:STATe]?	Y
<b><i>Frequency Modulation Subsystem</i></b>	
:FM:COUPling AC DC	Y
:FM:COUPling?	Y
:FM:SENSitivity <value><freqsuffix/V>	Y
:FM:SENSitivity?	Y
:FM:SOURce1 EXTernal :FM:SOURce EXTernal	N
:FM:SOURce1? :FM:SOURce?	N Y
:FM:STATe ON OFF 1 0	Y
:FM:STATe?	Y
<b><i>Frequency Subsystem</i></b>	
:FREQuency:CENTer <value>[<freqsuffix>] UP DOWN	Y
:FREQuency:CENTer?	Y
:FREQuency:MANual <value><unit> UP DOWN	N
[ :SOURce[1]]:FREQuency:MANual? [:SOURce]:FREQuency:MANual?	N Y
:FREQuency:MODE FIXed CW SWEep SWCW	N
:FREQuency:MODE?	Y
:FREQuency:MULTiplier <value>	Y
:FREQuency:MULTiplier:STATe ON OFF 1 0	N
:FREQuency:MULTiplier:STATe?	N
:FREQuency:MULTiplier?	Y
:FREQuency:OFFSet <value>	Y
:FREQuency:OFFSet:STATe ON OFF 1 0	Y
:FREQuency:OFFSet:STATe?	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:FREQuency:OFFSet?	Y
:FREQuency:SPAN <value>[<freqsuffix>] UP DOWN	Y
:FREQuency:SPAN?	Y
:FREQuency:START <value>[<freqsuffix>] UP DOWN	Y
:FREQuency:START?	Y
:FREQuency:STEP[:INCRement] <value>[<freqsuffix>]	Y
:FREQuency:STEP[:INCRement]?	Y
:FREQuency:STOP <value>[<freqsuffix>] UP DOWN	Y
:FREQuency:STOP?	Y
:FREQuency[:CW :FIXed] <value>[<freqsuffix>] UP DOWN	Y
:FREQuency[:CW :FIXed]:AUTO ON OFF 1 0	N
:FREQuency[:CW :FIXed]:AUTO?	N
:FREQuency[:CW :FIXed]?	Y
<b>Initiate Subsystem</b>	
:INITiate:CONTInuous ON OFF 1 0	Y
:INITiate:CONTInuous?	Y
:INITiate[:IMMediate]	Y
<b>Marker Subsystem</b>	
[:SOURce[1]]:MARKer[n]:AMPLitude[:STATe] ON OFF 1 0	N
[:SOURce]:MARKer[n]:AMPLitude[:STATe] ON OFF 1 0	Y
[:SOURce[1]]:MARKer[n]:AMPLitude[:STATe]?	N
[:SOURce]:MARKer[n]:AMPLitude[:STATe]?	Y
:MARKer[n]:AOFF	Y
:MARKer[n]:FREQuency <value><unit>	Y
:MARKer[n]:FREQuency?	N
:MARKer[n]:MODE FREQuency DELTA	Y



**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:MARKer[n]:MODE?	Y
:MARKer[n]:REFerence <n>	Y
:MARKer[n]:REFerence?	Y
:MARKer[n][:STATe] ON OFF 1 0	N
:MARKer[n][:STATe]?	N
<b>Memory Subsystem</b>	
:MEMory:RAM:INITialize[:ALL]	N
<b>Output Subsystem</b>	
:OUTPut:IMPedance?	N
:OUTPut[:STATe] ON OFF 1 0	Y
:OUTPut[:STATe]?	Y
<b>Power Subsystem</b>	
:POWer:ALC:CFACTOR <value>[DB] UP DOWN	Y
:POWer:ALC:CFACTOR?	Y
:POWer:ALC:SOURce1 INTernal DIODE PMETer MMHead :POWer:ALC:SOURce INTernal DIODE PMETer MMHead	N
:POWer:ALC:SOURce1? :POWer:ALC:SOURce?	N Y
:POWer:ALC[:STATe] ON OFF 1 0	Y
:POWer:ALC[:STATe]?	Y
:POWer:ATTenuation <value>[DB] UP DOWN	Y
:POWer:ATTenuation:AUTO ON OFF 1 0	Y
:POWer:ATTenuation:AUTO?	Y
:POWer:ATTenuation?	Y
:POWer:CENTer <value>[<lvlsuffix>] UP DOWN	Y
:POWer:CENTer?	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:POWer:MODE FIXed SWEep	Y
:POWer:MODE?	Y
:POWer:OFFSet <value>[DB] UP DOWN	Y
:POWer:OFFSet:STATe ON OFF 1 0	Y
:POWer:OFFSet:STATe?	Y
:POWer:OFFSet?	Y
:POWer:SLOPe <value>[DB/freqsuffix] UP DOWN	N
:POWer:SLOPe:STATe ON OFF 1 0	N
:POWer:SLOPe:STATe?	N
:POWer:SLOPe?	Y
:POWer:SPAN <value>[DB] UP DOWN	Y
:POWer:SPAN?	Y
:POWer:START <value>[<lvlsuffix>] UP DOWN	Y
:POWer:START?	Y
:POWer:STATe ON OFF 1 0	Y
:POWer:STATe?	Y
:POWer:STEP[: INCRement] <value>[DB]	Y
:POWer:STEP[: INCRement]?	Y
:POWer:STOP <value>[<lvlsuffix>] UP DOWN	Y
:POWer:STOP?	Y
:POWer[: LEVel] <value>[<lvlsuffix>] UP DOWN	Y
:POWer[: LEVel]?	Y
<b><i>Pulse Modulation Subsystem</i></b>	
:PULM:SOURce1 INTernal EXTernal SCALar SQ1K	N
:PULM:SOURce INTernal EXTernal SCALar SQ1K	

**Table 7-5 8375xB SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83751B &amp; 83752B</b>
:PULM:SOURcel?	N
:PULM:SOURce?	Y
:PULM:STATe ON OFF 1 0	Y
:PULM:STATe?	Y
<b><i>Pulse Subsystem</i></b>	
:PULSe:FREQuency <value>[<freqsuffix>]	Y
:PULSe:FREQuency?	Y
:PULSe:PERiod <value>[<timesuffix>]	Y
:PULSe:PERiod?	Y
:PULSe:WIDTh <value>[<timesuffix>]	Y
:PULSe:WIDTh?	Y
<b><i>Reference Oscillator Subsystem</i></b>	
:ROSCillator:SOURcel INTERNAL EXTERNAL NONE	N
:ROSCillator:SOURce INTERNAL EXTERNAL NONE	Y
:ROSCillator:SOURcel:AUTO ON OFF 1 0	N
:ROSCillator:SOURce:AUTO ON OFF 1 0	Y
:ROSCillator:SOURcel:AUTO?	N
:ROSCillator:SOURce:AUTO?	Y
:ROSCillator:SOURcel?	N
:ROSCillator:SOURce?	Y
<b><i>Status Subsystem</i></b>	
:STATus:OPERation:CONDition?	Y
:STATus:OPERation:ENABle <value>	Y
:STATus:OPERation:ENABle?	Y
:STATus:OPERation:NTRansition <value>	Y
:STATus:OPERation:NTRansition?	Y
:STATus:OPERation:PTRansition <value>	Y
:STATus:OPERation:PTRansition?	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:STATus:OPERation[:EVENT]?	Y
:STATus:PRESet	Y
:STATus:QUESTionable:CONDition?	Y
:STATus:QUESTionable:ENABle <value>	Y
:STATus:QUESTionable:ENABle?	Y
:STATus:QUESTionable:NTRansition <value>	Y
:STATus:QUESTionable:NTRansition?	Y
:STATus:QUESTionable:PTRansition <value>	Y
:STATus:QUESTionable:PTRansition?	Y
:STATus:QUESTionable[:EVENT]?	Y
<b><i>Sweep Subsystem</i></b>	
:SWEep:CONTRol:TYPE MASTER SLAVE	Y
:SWEep:CONTRol:TYPE?	Y
:SWEep:DWELl <value>[<timesuffix>]	Y
:SWEep:DWELl:AUTO ON OFF 1 0	N
:SWEep:DWELl:AUTO?	N
:SWEep:DWELl?	Y
:SWEep:GENERation ANALog STEPped	Y
:SWEep:GENERation?	Y
:SWEep:MANual:POINt <value>	Y
:SWEep:MANual:POINt?	Y
:SWEep:MANual[:RELative] <value>	N
:SWEep:MANual[:RELative]?	N
:SWEep:MARKer:STATe ON OFF 1 0	N
:SWEep:MARKer:STATe?	N

**Table 7-5 8375xB SCPI Commands (Continued)**

<b>Y= Supported by PSG N= Not supported by PSG</b>	<b>83751B &amp; 83752B</b>
:SWEep:MARKer:XFER	N
:SWEep:MODE AUTO MANual	Y
:SWEep:MODE?	Y
:SWEep:POINTs <value>	Y
:SWEep:POINTs?	Y
:SWEep:POWer:STEP <value>[<lvlsuffix>] UP DOWN	N
:SWEep:POWer:STEP?	N
:SWEep:TIME <value>[<timesuffix>]	N
:SWEep:TIME:AUTO ON OFF 1 0	N
:SWEep:TIME:AUTO?	Y
:SWEep:TIME:LLIMit <value>[<timesuffix>]	Y
:SWEep:TIME:LLIMit?	Y
:SWEep:TIME?	Y
:SWEep[:FREQuency]:STEP <value>[<freqsuffix>] UP DOWN	N
:SWEep[:FREQuency]:STEP?	N
:SWEep[:POINTs]:TRIGger:SOURce IMMEDIATE BUS EXTERNAL :SWEep[:POINTs]:TRIGger:SOURce IMMEDIATE BUS EXTERNAL	N
:SWEep[:POINTs]:TRIGger:SOURce? :SWEep[:POINTs]:TRIGger:SOURce?	N
:SWEep[:POINTs]:TRIGger[:IMMEDIATE]	N
<b><i>System Subsystem</i></b>	
:SYSTem:ALTerNate <reg num>	Y
:SYSTem:ALTerNate:STATe ON OFF 1 0	Y
:SYSTem:ALTerNate:STATe?	Y
:SYSTem:ALTerNate?	Y
:SYSTem:COMMunicate:GPIB:ADDRESS <value>	Y

**Table 7-5 8375xB SCPI Commands (Continued)**

<b>Y= Supported by PSG</b> <b>N= Not supported by PSG</b>	<b>83751B &amp;</b> <b>83752B</b>
:SYSTem:COMMunicate:PMETer:ADDRess <value>	Y
:SYSTem:COMMunicate:PMETer:ADDRess?	Y
:SYSTem:COMMunicate:PMETer:TYPE SCPI 70100A 437B 438A	N
:SYSTem:COMMunicate:PMETer:TYPE?	N
:SYSTem:ERRor?	Y
:SYSTem:KEY:DISAbLe SAVE	N
:SYSTem:KEY:DISAbLe? SAVE	N
:SYSTem:KEY:ENABLe SAVE	N
:SYSTem:KEY:ENABLe? SAVE	N
:SYSTem:KEY[:CODE] <value>	N
:SYSTem:KEY[:CODE]?	N
:SYSTem:LANGUage "SCPI" "TMSL" "COMP"	N
:SYSTem:LANGUage?	Y
:SYSTem:PRESet:TYPE FACTory USER	Y
:SYSTem:PRESet:TYPE?	Y
:SYSTem:PRESet[:EXECute]	Y
:SYSTem:PRESet[:USER]:SAVE	Y
:SYSTem:SECurity:CLEar	N
:SYSTem:SECurity:COUNt <value>	Y
:SYSTem:SECurity:KLOCK ON OFF 0 1	N
:SYSTem:SECurity:ZERO ON OFF 0 1	N
:SYSTem:VERSion?	Y
<b>Trigger Subsystem</b>	
:TRIGger:SOURce1 IMMEDIATE BUS EXTERNAL HOLD :TRIGger:SOURce IMMEDIATE BUS EXTERNAL HOLD	N

**Table 7-5 8375xB SCPI Commands (Continued)**

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:TRIGger:SOURcel?	N
:TRIGger:SOURce?	Y
:TRIGger[:IMMediate]	Y
<b><i>Tsweep Subsystem</i></b>	
:TSWEEP	Y

## 8662A/63A Compatible Commands

The tables in this section provide the following:

**Table 7-6 on page 400:** a comprehensive list of 8662A/63A programming commands, listed in alphabetical order. The equivalent SCPI command sequence for each supported code is provided. Codes that have no equivalent SCPI command sequence are indicated in the command column, as are codes that are *not* supported by the PSG family.

**Table 7-7 on page 406:** a list of the implemented 8662B/63B programming commands that set the active function. This table also indicates which codes are compatible with the increment (up), and the decrement (down) SCPI commands.

To use the commands, select **866xA** as the remote language. See “:LANGUage” on page 93 for selecting the language type.

When using the programming codes in this section, you can:

- set the PSG system language to 866xA for the current session:  
**Utility > GPIB/RS-232 LAN > Remote Language > 866xA**  
 or send the command:  
 :SYST:LANG "8662" or "8663"
- set the PSG system language to 866xA so that it does not reset on a preset, an instrument power cycle or a \*RST command:  
**Utility > Power On/Preset > Preset Language > 866xA**  
 or send the command:  
 :SYST:PRESET:LANG "8662" or "8663"
- set the \*IDN? response to any 866xA-like response you prefer. Refer to the :SYSTEM:IDN command on page 347.

---

**NOTE** Compatibility is provided for GPIB only; RS-232 and LAN are *not* supported.  
 Device Clear does not preset the instrument.  
 To reproduce the sweep functionality, use the PSG List Sweep features.

---

**Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences**

Command	Description	8662	8663	Equivalent SCPI Command Sequence
@1	Write require service mask	Y	Y	*SRE <mask> <i>supported, however, only 4 of the 8 bits of the 8662/8663 status byte will be set. These are: Entry Error, Hardware Error, Power Fail, and Request Service Message (SRQ).</i>
@2	Deferred execution mode	N	N	<i>not supported</i>
@3	Immediate execution mode	N	N	<i>not supported</i>
+D	+dBm	Y	Y	DEM
AM	AM modulation <i>See also: Table 7-7 on page 406</i>	Y		AM:DEPTH <val> <units> AM:TRAC ON FM:STAT OFF AM:STAT ON
			Y	AM:DEPTH <val> <units> AM:TRAC ON AM:STAT ON
AO	Amplitude off	Y	Y	OUTPut:STATe OFF
AP	Amplitude	Y	Y	POW:REF:STATe OFF POWer:AMPL <val> <units> OUTPut:STATe ON <i>See also: Table 7-7 on page 406</i>
AS BLSQ	Auto sequence	N	N	<i>not supported</i>
BP	BPSK modulation		N	<i>not supported</i>
CT	Configure trigger	Y	Y	<i>no equivalent SCPI command sequence</i>
-D	-dBm Negates the power value.	Y	Y	DEM
DB	dB	Y	Y	DB
DG	Degree	Y		DEG
DM	dBm	Y	Y	DEM
DN	Decrement Passes DOWN as parameter of active function command.	Y	Y	<i>See Table 7-7 on page 406</i>
FA	Start frequency	Y	Y	<i>See W2, W3, W4, and Table 7-7 on page 406</i>
FB	Stop frequency	Y	Y	<i>See W2, W3, W4, and Table 7-7 on page 406</i>
FM	FM modulation <i>See also: Table 7-7 on page 406</i>	Y		FM:DEV <val> <units> AM:STAT OFF FM:STAT ON
			Y	FM:DEV <val> <units> FM:STAT ON



**Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)**

Command	Description	8662	8663	Equivalent SCPI Command Sequence
FR	Center frequency	Y	Y	FREquency:CW <val> <units> <i>See also: W2, W3, and W4, and Table 7-7 on page 406</i>
FS	Span frequency	Y	Y	<i>See W2, W3, W4, and Table 7-7 on page 406</i>
GZ	GHz	Y	Y	GHZ
HZ	Hz	Y	Y	HZ
IS	Set increment Adds STEP:INCR to active function command.	Y	Y	<i>no equivalent SCPI command sequence</i>
KZ	kHz	Y	Y	KHZ
L1	Learn front panel	N	N	<i>not supported</i>
L2	Fast learn	N	N	<i>not supported</i>
MO M0	Modulation off	Y	Y	AM:STATe OFF FM:STATe OFF PULM:STATe OFF PM:STATe OFF
M1	Modulation source internal 400 Hz  <b>For 8663A:</b> Executes MF with <freq> = 400 Hz	Y		<mod>:SOURce INT1 <mod>:INT1:FREQ 400Hz
			Y	AM:INT1:FREQ 400 MHz FM:INT2:FREQ 400 MHz PM:INT2:FREQ 400 MHz PULM:INT:FREQ 400 MHz
M2	Modulation source internal 1 kHz  <b>For 8663A:</b> Executes MF with <freq> = 1 kHz	Y		<mod>:SOURce INT1 <mod>:INT1:FREQ 1kHz
			Y	AM:INT1:FREQ 1 kHz FM:INT2:FREQ 1 kHz PM:INT2:FREQ 1 kHz PULM:INT:FREQ 1 kHz
M3	Modulation source external AC  <b>For 8663A:</b> <mod> = AM, FM, or PM, depending on which is on. <n> = 1 for AM, 2 for FM or PM <b>NOTE:</b> For PM, the impedance value is set using the SP71/SP70 commands	Y		<mod>:SOURce EXT <mod>:EXT:COUPling AC <mod>:EXT:IMP 600
			Y	<mod>:SOURce EXT<n> <mod>:EXT<n>:COUPling AC <mod>:EXT<n>:IMP 600
M4	Modulation source external DC  <b>For 8663A:</b> <mod> = AM, FM, or PM, depending on which is on. <n> = 1 for AM, 2 for FM or PM <b>NOTE:</b> For PM, the impedance value is set using the SP71/SP70 commands	Y		<mod>:SOURce EXT <mod>:EXT:COUPling DC <mod>:EXT:IMP 600
			Y	<mod>:SOURce EXT<n> <mod>:EXT<n>:COUPling DC <mod>:EXT<n>:IMP 600

**Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)**

Command	Description	8662	8663	Equivalent SCPI Command Sequence
MF	Modulation frequency  <mod> = FM, or PM, depending on which is on.  <i>Also see: M1, M2, and Table 7-7 on page 406</i>		Y	<b>AM:</b> AM:SOUR:INT1 AM:SOUR:INT1:FREQ <freq> <b>FM or PM:</b> <mod>:SOUR:INT2 <mod>:SOUR:INT2:FREQ <freq> <b>Pulse:</b> PULM:SOUR:INT PULM:INT:FREQ <freq> PULM:SOUR:INT SQUARE
MS	Read status key message Returns status string.	Y	Y	<i>no equivalent SCPI command sequence</i>
MV	mV	Y	Y	MV
MZ	MHz	Y	Y	MHZ
N1	Linear 100 steps	Y	Y	<i>See W2, W3, and W4</i>
N2	Linear 1000 steps	Y	Y	<i>See W2, W3, and W4</i>
N3	Step size	Y	Y	<i>See W2, W3, W4, and Table 7-7 on page 406</i>
N4	Log 10% steps	Y	Y	<i>See W2, W3, and W4</i>
N5	Log 1% steps	Y	Y	<i>See W2, W3, and W4</i>
PC	%	Y	Y	PCT
PL	Pulse modulation Must have an instrument with pulse capability.		Y	PULM:STAT ON
PM	Phase modulation Not compatible with any FM modulation.		Y	PM:STAT ON <i>See also: Table 7-7 on page 406</i>
R1	Knob resolution x10	N	N	<i>not supported</i>
R2	Knob resolution /10	N	N	<i>not supported</i>
R3	Knob off	N	N	<i>not supported</i>
R4 BLR1	Knob hold	N	N	<i>not supported</i>
R5 BLR2	Knob increment	N	N	<i>not supported</i>
RC	Recall	Y	Y	*RCL
RD	Knob down Only for manual sweep	Y	Y	LIST:MANual DOWN
RM	Read require service mask	N	N	*SRE? <i>supported, however, only 4 of the 8 bits of the 8662/8663 status byte will be set. These are: Entry Error, Hardware Error, Power Fail, and Request Service Message (SRQ).</i>
RU	Knob up Only for manual sweep	Y	Y	LIST:MANual UP

Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
SP00	System preset Presets the instrument, including the compatibility language.	Y	Y	SYSTEM:PRESet
SP10	Frequency offset off	Y	Y	FREQ:OFFS:STAT OFF
SP11	Positive frequency offset The 8662 modifies the output, but does not change the displayed frequency; the PSG modifies the displayed frequency, but does <i>not</i> change the output. Because of this, you must first set the offset, then reapply the frequency to change the output.	Y	Y	FREQ:OFFS <value> FREQ:OFFS:STAT ON FREQ:CW <displayed value>
SP12	Negative frequency offset The 8662 modifies the output, but does not change the displayed frequency; the PSG modifies the displayed frequency, but does <i>not</i> change the output. Because of this, you must first set the offset, then reapply the frequency to change the output.	Y	Y	FREQ:OFFS <value> FREQ:OFFS:STAT ON FREQ:CW <displayed value>
SP20	ALC bandwidth normal		Y	POW:ALC:BANDwidth:AUTO ON
SP21	ALC bandwidth < 1 kHz		Y	Power:ALC:BANDwidth:AUTO OFFPOWER:ALC:BANDwidth 1KHZ
SP30	Amplitude reference off	Y	Y	POW:REF:STATe OFF
SP31	Amplitude reference	Y	Y	POW:REF <val> <val> = current amplitude setting POW:REF:STATe ON
SP32	Amplitude reference relative to 1 $\mu$ V		Y	POW:REF 106.99DBM POW:REF:STATe ON POW 1UV
SP40	External AM off	Y		AM:STAT OFF
	Modulation frequency sweep mode off		N	<i>not supported</i>
SP41	Internal FM + external AM (AC)	Y		FM:SOUR INT1 FM:INT1:FREQ 400 HZ FM:STAT ON AM:SOUR EXTL AM:EXT1:IMP 600 AM:DEPTH 95 PCT AM:EXT1:COUP AC AM:STAT ON
	Modulation frequency sweep mode on		N	<i>not supported</i>
SP42	Internal FM + external AM (DC)	Y		FM:SOUR INT1 FM:INT1:FREQ 400 HZ FM:STAT ON AM:SOUR EXTL AM:EXT1:IMP 600 AM:DEPTH 95 PCT AM:EXT1:COUP DC AM:STAT ON
SP50	AUX FM off	Y	Y	FM2:STAT OFF

Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
SP51	AUX FM on			
	<b>RF (MHz) FM Deviation (kHz)</b>			
	0.01–120 25			
	120–160 6.25			
	160–320 12.5			
	320–640 25			
	640–1280 50			
	1280–2560 100			
	<dev> is dependant on output frequency, and mimics the 8662 hardware settings.	Y	Y	FM2:SOUR EXT2 FM2:EXT2:COUP DC FM2:EXT2:IMP 600 FM2:DEV <dev> kHz FM2:STAT ON
SP60	Parameter shift keying off	N	N	<i>not supported</i>
SP61	Parameter shift keying up/down (two-key)	N	N	<i>not supported</i>
SP62	Parameter shift keying up/down (one-key)	N	N	<i>not supported</i>
SP70	External PM input impedance 50Ω Effects the behavior of M3 and M4.		Y	<i>no equivalent SCPI command sequence</i>
SP71	External PM input impedance 600Ω Effects the behavior of M3 and M4.		Y	<i>no equivalent SCPI command sequence</i>
SP80	Special functions 10-62 off	Y	Y	FM2:STAT OFF AM:STAT OFF FREQ:OPFS:STAT OFF
SP81	Amplitude conversion (V-dBm)	N	N	<i>not supported</i>
SP82	Display GPIB address	N	N	<i>not supported</i>
SP83	ROM test	N	N	<i>not supported</i>
SP84	RAM test	N	N	<i>not supported</i>
SP85	Amplitude correction off	Y	Y	POWER:ALC:STATe OFF
SP86	Amplitude correction on PSG ALC ON always works with sweep.	Y	Y	POWER:ALC:STATe ON
SP87	Amplitude correction on (includes Sweep)		Y	POWER:ALC:STATe ON
SP87	GPIB operator request response	N		<i>not supported</i>
SP88	Auto sequence	N	N	<i>not supported</i>
SP89	GPIB operator request response		N	<i>not supported</i>
SP90	Set auto sequence step delay		N	<i>not supported</i>
SP91	Enable frequency hopping mode		N	<i>not supported</i>
SP92	Knob (restore normal operation)		N	<i>not supported</i>
SP93	Manual amplitude level control		N	<i>not supported</i>
SP94	Knob, 120 increments per revolution		N	<i>not supported</i>
SP95	Knob, 120 increments per revolution, reconfigure AUX con.		N	<i>not supported</i>

**Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)**

Command	Description	8662	8663	Equivalent SCPI Command Sequence
SP96	Modulation oscillator off when modulation is off		N	<i>not supported</i>
SP97	Modulation oscillator on		N	<i>not supported</i>
SP98	Turn display on		Y	DISP ON
SP99	Turn display off		Y	DISP OFF
SP2.0	Power up preset off		N	<i>not supported</i>
SP2.1	Power up preset on		N	<i>not supported</i>
SQ	Sequence	N	N	<i>not supported</i>
SS BLST	Set sequence	N	N	<i>not supported</i>
ST	Store Saves/recalls register to sequence 0.	Y	Y	*SAV
T1	0.5 ms per step	Y	Y	SWEEP:DWELL 0.5ms <i>Beyond PSG range limit; is set to 1ms.</i>
T2	1 ms per step	Y	Y	SWEEP:DWELL 1ms
T3	2 ms per step	Y	Y	SWEEP:DWELL 2ms
T4	10 ms per step	Y	Y	SWEEP:DWELL 10ms
T5	100 ms per step	Y	Y	SWEEP:DWELL 100ms
TR	Trigger Performs command code setup with CT command.	Y	Y	<i>no equivalent SCPI command sequence</i>
UP	Increment Passes UP as a parameter of the active function command.	Y	Y	<a href="#">See Table 7-7 on page 406</a>
UV	mV	Y	Y	UV
W1	Sweep off	Y	Y	FREQ:MODE CW LIST:TRIG:SOUR IMM
W2	Auto sweep mode on  Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5 <b>Default values:</b> FR = 100 MHz, FS = 10 MHz, N1, T2 FA = 1 MHz, FB = 1279 MHz	Y	Y	INIT:CONT ON SWEEP:MODE AUTO LIST:TRIG:SOUR IMM LIST:DWELL:TYPE STEP LIST:TYPE LIST FREQ:MODE LIST
W3	Manual sweep mode on  Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5 <b>Default values:</b> FR = 100 MHz, FS = 10 MHz, N1, T2 FA = 1 MHz, FB = 1279 MHz	Y	Y	INIT:CONT ON SWEEP:MODE MANuaL LIST:TRIG:SOUR IMM LIST:DWELL:TYPE STEP LIST:TYPE LIST FREQ:MODE LIST

**Table 7-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)**

Command	Description	8662	8663	Equivalent SCPI Command Sequence
W4	Single sweep mode on  Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5 <b>Default values:</b> FR = 100 MHz, FS = 10 MHz, N1, T2 FA = 1 MHz, FB = 1279 MHz	Y	Y	INIT:CONT OFF SWEEP:MODE AUTO LIST:TRIG:SOUR IMM LIST:DWELL:TYPE STEP LIST:TYPE LIST FREQ:MODE LIST INIT
X1	Marker 1	N	N	<i>not supported</i>
X2	Marker 2	N	N	<i>not supported</i>
X3	Marker 3	N	N	<i>not supported</i>
X4	Marker 4	N	N	<i>not supported</i>
X5	Marker 5	N	N	<i>not supported</i>
X6	Marker off	N	N	<i>not supported</i>
X7 BLX6	All markers off	N	N	<i>not supported</i>
Y0	Remote stepped sweep off	Y	Y	FREQ:MODE CW LIST:TRIG:SOUR IMM
Y1 Y2	Remote stepped sweep on	Y	Y	INIT:CONT ON SWEEP:MODE AUTO LIST:DWELL:TYPE STEP LIST:TYPE LIST FREQ:MODE LIST LIST:TRIG:SOUR BUS
Y3	Execute remote stepped sweep	Y	Y	*TRG

**Table 7-7 8662/63B Command Compatibility**

Command	Description	Sets Active Function	Compatible with UP/DN	8662	8663	Equivalent SCPI Commands for UP/DN and Increment
AM	AM modulation	Y	Y	Y	Y	AM:DEPTH UP AM:DEPTH DOWN AM:DEPTH:STEP:INCR
AP	Amplitude	Y	Y	Y	Y	POW:AMPL UP POW:AMPL DOWN POW:AMPL:STEP:INCR
FA	Start frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
FB	Stop frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
FM	FM modulation	Y	Y	Y	Y	FM:DEV UP FM:DEV DOWN FM:DEV:STEP:INCR
FR	Center frequency	Y	Y	Y	Y	FREQ:CW UP FREQ:CW DOWN FREQ:CW:STEP:INCR

**Table 7-7 8662/63B Command Compatibility (Continued)**

Command	Description	Sets Active Function	Compatible with UP/DN	8662	8663	Equivalent SCPI Commands for UP/DN and Increment
FS	Span frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
MF	Modulation frequency	Y	Y		Y	<mod>:INT:FREQ UP <mod>:INT:FREQ DOWN <mod>:INT:FREQ:STEP:INCR <mod> = AM FM PM PULM
N3	Step size	Y	Y	Y	Y	<i>no equivalent SCPI commands</i>
PM	Phase modulation Not compatible with any FM modulation.	Y	Y		Y	PM:DEV UP PM:DEV DOWN PM:DEV:STEP:INCR





**Symbols**

◊M Dev Couple Off On softkey [189](#)  
 ◊M Dev softkey [188](#)  
 ◊M Off On softkey [187](#)  
 ◊M Path 1 2 softkey [181](#)  
 ◊M Stop Rate softkey [183](#)  
 ◊M Sweep Time softkey [186](#)  
 ◊M Tone 2 Ampl Percent of Peak softkey [185](#)  
 $\pi/4$  DQPSK softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 # of Carriers softkey [287, 290](#)  
 # Points softkey [137](#)  
 # Skipped Points softkey [256](#)

**Numerics**

1048576 softkey [203](#)  
 128QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 131072 softkey [203](#)  
 16 1's & 16 0's softkey  
     *See* custom subsystem keys  
 16384 softkey [203](#)  
 16PSK softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 16QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 2.100 MHz softkey [200](#)  
 256QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 262144 softkey [203](#)  
 2-Lvl FSK softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 32 1's & 32 0's softkey  
     *See* custom subsystem keys  
 32768 softkey [203](#)  
 32QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 4 1's & 4 0's softkey  
     *See* custom subsystem keys  
 40.000 MHz softkey [198, 200](#)  
     digital modulation subsystem [231, 243, 251](#)  
     external I/Q filter [276, 298, 322](#)  
     I/Q modulation filter [253, 280, 324](#)  
     modulation attenuation [300](#)  
 4-Lvl FSK softkey  
     *See* custom subsystem keys

*See* Dmodulation subsystem keys  
 4QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 524288 softkeys [203](#)  
 64 1's & 64 0's softkey  
     *See* custom subsystem keys  
 64QAM softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys  
 65536 softkey [203](#)  
 8 1's & 8 0's softkey  
     *See* custom subsystem keys  
 8340, 8360, 8757 Language [97](#)  
 8340B/41B, compatible commands [350](#)  
 836xxB/L, compatible commands [363](#)  
 8371xB, compatible commands [379](#)  
 8373xB, compatible commands [379](#)  
 8375xB, compatible commands [387](#)  
 8648A/B/C/D softkey [93, 97](#)  
 8656B,8657A/B softkey [93, 97](#)  
 8657D NADC softkey [93, 97](#)  
 8657D PDC softkey [93, 97](#)  
 8657J PHS softkey [93, 97](#)  
 8662A/63A, compatible commands [399](#)  
 8757D, compatible commands [350](#)  
 8PSK softkey  
     *See* custom subsystem keys  
     *See* Dmodulation subsystem keys

**A**

abort list/step sweep [105](#)  
 Access denied [53](#)  
 Activate Secure Display softkey [99](#)  
 Add Comment To Seq[n] Reg[nn] softkey [58](#)  
 Adjust Phase softkey [125](#)  
 ALC [142, 147](#)  
 ALC BW softkey [142](#)  
 ALC Hold [258, 281, 325](#)  
 ALC hold markers  
     Dmodulation subsystem [281](#)  
     dual ARB subsystem [258](#)  
     multitone subsystem [301](#)  
     two tone subsystem [325](#)  
 ALC level [143](#)  
 ALC Off On softkey [147](#)  
 Align DACs softkey [214, 248](#)  
 Alignment Left Cent Right softkey [321](#)  
 All softkey [43, 57](#)  
 alternate amplitude markers  
     AWGN ARB subsystem [200](#)  
 alternate frequency [170, 177](#)  
 Alternate Sweep Off On softkey [90](#)

---

# Index

- Alternate Sweep softkey [90](#)
  - AM softkeys
    - AM Depth [159](#), [165](#), [166](#), [158](#), [164](#), [157](#), [159](#), [161](#), [163](#), [161](#), [162](#), [161](#), [165](#)
  - Ampl softkeys
    - Ampl Offset [153](#), [151](#), [152](#)
  - amplitude
    - LF output [176](#)
    - list sweep points [132](#)
  - amplitude and frequency correction pair [110](#)
  - Amplitude hardkey [153](#), [154](#)
  - Amplitude Markers Off On softkey [139](#)
  - amplitude modulation subsystem keys
    - AM Depth [159](#), [165](#), [166](#), [158](#), [164](#), [157](#), [159](#), [161](#), [163](#), [161](#), [162](#), [161](#), [165](#)
    - Ext Coupling DC AC [160](#), [164](#)
    - Gaussian [162](#), [184](#)
    - Incr Set [158](#), [167](#), [164](#)
    - Negative [162](#), [184](#)
    - Positive [162](#), [184](#)
    - Uniform [162](#), [184](#)
  - amplitude units [30](#)
  - APCO 25 [223](#)
  - APCO 25 C4FM softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
    - See* GPS subsystem keys
  - APCO 25 w/C4FM softkey [223](#), [287](#), [289](#)
  - APCO 25 w/C4QPSK softkey [287](#), [289](#)
  - APCO 25 w/CQPSK softkey [223](#)
  - Apply Settings softkey [321](#)
  - Apply to Waveform softkey [254](#), [256](#)
  - Arb AWGN Off On softkey [205](#)
  - ARB Off On softkey [275](#)
  - arb player [11](#)
  - ARB Reference Ext Int softkey
    - See* AWGN subsystem keys
    - See* Dmodulation subsystem keys
    - See* dual ARB subsystem keys
    - See* multitone subsystem keys
  - ARB sample clock rate [261](#)
  - ARB Sample Clock softkey [204](#), [265](#), [286](#), [305](#), [329](#)
  - arbitrary waveform
    - clipping [248](#)
    - runtime scaling [264](#)
    - scaling files [265](#)
  - Atten Hold Off On softkey [148](#)
  - attenuator [28](#), [236](#), [241](#), [242](#), [243](#), [279](#), [299](#), [323](#)
  - attenuator auto [241](#)
  - attenuator bandwidth [243](#)
  - automatic leveling control [142](#), [147](#)
  - AWGN
    - carrier bandwidth [262](#)
    - flat noise bandwidth [261](#)
    - noise state off on [261](#)
  - AWGN ARB subsystem keys
    - Marker Polarity Neg Pos [203](#)
  - AWGN subsystem
    - ALC hold [201](#)
    - RF blanking/pulse [202](#)
  - AWGN subsystem keys
    - 1048576 [203](#)
    - 2.100 MHz [200](#), [203](#)
    - 32768 [203](#)
    - 40.000 MHz [198](#), [200](#)
    - 524288 [203](#)
    - 65536 [203](#)
    - alc hold [201](#), [200](#)
    - Arb AWGN Off On [205](#), [204](#)
    - Bandwidth [198](#)
    - Clear Header [199](#)
    - I/Q Mod Filter Manual Auto [200](#), [198](#)
    - Modulator Atten Manual Auto [199](#)
    - Noise Seed Fixed Random [205](#), [202](#)
    - Reference Freq [203](#)
    - Save Setup To Header [199](#)
    - Through [198](#), [200](#)
    - Waveform Length [203](#)
- ## B
- backward compatible SCPI commands
    - \*IDN? output [347](#)
    - 8340B/41B [350](#), [363](#), [379](#), [387](#), [399](#), [350](#)
  - band and channel selection [117](#)
  - Bandwidth softkey [198](#), [205](#)
  - Baseband Frequency Offset softkey
    - Dual ARB subsystem keys [247](#)
  - baud rate [26](#)
  - BBG Data Clock Ext Int softkey
    - See* custom subsystem keys
  - BBG Ref Ext Int softkey
    - See* custom subsystem keys
  - BBG1 softkey [233](#), [246](#)
  - Binary softkey [39](#), [59](#)
  - binary values [13](#)
  - Bit softkey [39](#)
  - blanking [65](#)
  - blanking, display [33](#)
  - Bluetooth softkey [223](#)
  - boolean SCPI parameters [7](#)
  - boolean, numeric response data [8](#)
  - BPSK softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
  - Brightness softkey [31](#)
  - Build New Waveform Sequence softkey [265](#)
  - burst

- rise time 212
- shape 52, 213, 211, 212
- Burst Gate In Polarity Neg Pos softkey 69, 70
- Bus softkey
  - AM trigger source 163
  - Dmodulation subsystem keys 294
  - dual ARB subsystem keys 271
  - FM trigger source 171
  - list trigger source 133, 179
  - modulation subsystem keys 186
  - trigger subsystem keys 106
- bus trigger source
  - custom subsystem 227
  - Dmodulation subsystem 294
  - dual ARB subsystem 271
- C**
- calibration subsystem 16
- calibration subsystem keys 18
  - Calibration Type DC User Full 17, 20
  - DCFM/DCΦM Cal 16
  - Execute Cal 16, 19
  - I/Q Calibration 16
  - Revert to Default Cal Settings 17, 19
  - Start Frequency 18, 20, 18, 20
- Calibration Type DC User Full softkey 17
- capture screen 32
- carrier bandwidth 262
- Carrier Phases Fixed Random softkey 288
- carrier to noise 262
- catalog, mass memory subsystem 59
- CDPD softkey 223, 287, 289
- channel and band selection 117
- channel number 116
- channels 114
- clear header 62
- Clear Header softkey 199, 249, 279, 298, 322
- clearing markers 254, 255
- Clip |I+jQ| To softkey 248
- Clip |I| To softkey 248
- Clip |Q| To softkey 248
- clipping
  - waveform files 248
- Clipping softkey 248
- Clipping Type |I+jQ| |I|,|Q| softkey 248
- clock 31
- command tree, SCPI 5
- Common Mode I/Q Offset softkey 234
- communication subsystem keys
  - Default Gateway 22
  - GPIB Address 21
  - Hostname 22
  - IP Address 23
- LAN Config 22
- Meter Address 24, 25
- Power Meter 25
- Reset RS-232 26, 27
- subnet 23
- COMP,8340, 8360, 8757 Language 93
- compatible commands
  - 8257D/67D/E8663D 348
- Configure Cal Array softkey 110
- connector selection, triggering
  - custom subsystem 229
  - Dmodulation subsystem 295
  - dual ARB subsystem 272
- continuous
  - segment advance 270
- Continuous softkey
  - custom subsystem keys 224
  - Dmodulation subsystem keys 291
  - dual ARB subsystem keys 270
- continuous sweep 105
- continuous trigger
  - response selection
    - custom subsystem 225
    - Dmodulation subsystem 292
    - dual ARB subsystem 269
  - trigger mode
    - custom subsystem 224
    - Dmodulation subsystem 291
    - dual ARB subsystem 267
- contrast hardkeys 32
- control additive noise 262
- Copy File softkey 44, 60
- correction
  - frequency and amplitude pair 110
- correction subsystem 110
- correction subsystem keys
  - Configure Cal Array 110
  - Flatness Off On 112
  - Load From Selected File
    - flatness 110
  - Preset List 111
  - Store To File 111
- creating a waveform
  - multitone 298
  - sequence, dual ARB 265
- custom
  - continuous 224
  - gate 224
  - trigger 225
- Custom Digital Mod State softkey 287, 289
- Custom Off On softkey 230
- custom subsystem 230
  - delay query 217
  - predefined setup 223

---

# Index

triggering, *See* triggers  
custom subsystem keys  
     $\pi/4$  DQPSK 221  
    128AM 221, 214, 221  
    256QAM 221  
    32 1's & 32 0's 214, 221  
    4 1's & 4 0's 214, 221  
    64 1's & 64 0's 214, 221  
    8 1's & 8 0's 214, 221  
    Align DACs 214, 218, 223  
    BBG Data Clock Ext Int 206, 217, 223, 221, 210, 211, 227  
    CDPD 223, 224  
    D8PSK 221, 215  
    Ext 214, 227, 218, 217, 228, 229  
    Fall Delay 208, 209, 206, 207, 214, 215, 225, 219  
    Gate Active Low High 226, 224, 218, 221  
    I/Q Scaling 219, 221  
    MSK 221  
    None 223, 218  
    Optimize FIR For EVM ACP 213, 221  
    Patt Trig In 1, Patt Trig In 2 229, 220, 221, 214  
    QPSK 221  
    Rectangle 218, 211, 212, 218  
    Sine 213, 224, 222  
    Trigger & Run 225, 227  
    UN3/4 GSM Gaussian 218, 213, 214, 218, 220, 221, 220, 221  
CW frequency 124

## D

D8PSK softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys  
data  
    memory subsystem 44, 60  
data append  
    memory subsystem 45  
data bit 46  
data block 52  
Data Clock Out Neg Pos softkey 72  
Data Clock Polarity Neg Pos softkey 69, 71, 73  
data files 44, 60  
data FSK 47  
data IQ 49  
Data Mode Raw Enc TLM softkey 311, 312  
Data Out Polarity Neg Pos softkey 72, 74  
data pattern 214  
Data Polarity Neg Pos softkey 69, 71  
DATA/CLK/SYNC Rear Outputs Off On softkey 74  
date format 30  
dBm softkey 108  
dBuV softkey 108  
dBuVemf softkey 108  
DC softkey 177

DCFM/DC $\Phi$ M Cal softkey 16  
decimal values 13  
DECT softkey 287, 289  
default calibration 19  
Default Gateway softkey 22  
defaults, restore factory 126  
delay query 217  
delay, I/Q 233  
delay, triggering  
    custom subsystem 228  
    Dmodulation subsystem 295, 296  
    dual ARB subsystem 272, 273  
Delete softkeys  
    Delete All ARB DMOD Files 55, 56, 54, 55, 61, 56, 57, 61, 348, 57, 61  
Delta Markers softkey 140  
Delta Ref Set softkey 141  
deviation, FSK 283  
DHCP 22  
Diagnostic Info softkey 28, 29, 35, 92  
diagnostic subsystem keys  
    Diagnostic Info 28, 29  
    Installed Board Info 28  
    License Info 28  
    Options Info 29  
Diff. Data Encode Off On softkey 215  
Diff. Mode I Offset softkey 234  
Diff. Mode Q Offset softkey 235  
Digital Modulation Off On softkey 297  
digital modulation subsystem keys  
    40.000 MHz 231, 243  
    BBG1 233, 246  
    Common Mode I/Q Offset 234  
    Diff. Mode I Offset 234, 235  
    Ext 50 Ohm 233, 246, 233, 246, 236, 237, 242  
    High Crest Mode Off On 232  
    I Offset 238, 241, 233, 237, 244, 247, 331, 235, 236, 231, 239, 240, 245, 232, 244  
    Modulation Atten Optimize Bandwidth 243, 241, 242, 243  
    Off 233, 246  
    Q Offset 238, 239  
    Summing Ratio (SRC1/SRC2) x.xx dB 246  
    Through 231, 243  
digital signal interface module  
    *See* digital subsystem keys  
digital subsystem keys  
    Bit Order 337  
    Clock Phase 334, 335, 336  
    Data Type 344, 337  
    Frame Polarity 340  
    I Gain 338, 339, 340  
    Logic Type 345, 344  
    N5102A Off On 346, 338, 341, 339  
    Pass Through Preset 346, 345

- Q Gain 341, 342
  - Reference Frequency 335, 342
  - Scaling 342, 343, 339
  - Word Alignment 336, 343
  - directories 11
  - directory structure 11
  - discrete response data 8
  - discrete SCPI parameters 7
  - display 28
    - secure mode 99
  - display blanking 33
  - display subsystem keys
    - Brightness 31
    - display contrast 32
    - Inverse Video Off On 32
    - Update in Remote Off On 33
  - displayed amplitude units 30
  - DMOD softkey 40
  - Dmodulation subsystem
    - markers, *See* markers
    - triggering, *See* triggers
  - Dmodulation subsystem keys
    - $\pi/4$  DQPSK 284
    - # of Carriers 287, 290
    - 128QAM 284
    - 256QAM 284
    - 32QAM 284
    - 40.000 MHz 276, 280, 284
    - 64QAM 284
    - 8PSK 284
    - APCO 25 C4FM 277, 287, 289, 287, 289, 285, 286, 305, 329
    - BPSK 284, 294
    - Carrier Phases Fixed Random 288, 287, 289, 279, 291, 287, 289
    - D8PSK 284, 287, 289, 297
    - EDGE 287, 289, 294, 296, 295, 296
    - Filter Alpha 277, 278, 292, 283, 287
    - Gate Active Low High 293, 291, 277, 284, 287, 289
    - I/Q Mod Filter Manual Auto 280, 276, 286, 289, 284
    - Load/Store 288
    - Marker Polarity Neg Pos 284, 281, 282, 279, 284, 287
    - NADC 287, 289, 281, 282, 277
    - Off 286, 278, 284
    - Patt Trig In 1,2 295, 287, 289, 287, 289, 287, 289
    - QPSK 284
    - Rectangle 277, 204, 285, 292, 277
    - Save Setup To Header 279, 287, 291, 290
    - TETRA 287, 289, 276, 280, 292, 294
    - UN3/4 GSM Gaussian 277, 284
  - Do External Input Level Measurement softkey 243
  - Do Power Search softkey 143, 144, 145, 146
  - documentation, list of xxiii
  - Doppler Shift softkey 312
  - downloading files 53
  - dual ARB subsystem 248
    - alternate amplitude 258
    - clipping 248
    - generate sine 249
    - markers, *See* markers
    - runtime scaling 264
    - scaling waveform files 265
    - Through 251
    - triggering, *See* triggers
    - VCO clock 274
  - Dual ARB subsystem keys
    - setting baseband frequency offset value 247
  - dual ARB subsystem keys
    - # Skipped Points 256
    - 40.000 MHz 251, 253
    - Apply to Waveform 254, 256, 275, 263, 265
    - Build New Waveform Sequence 265, 271
    - carrier bandwidth 262
    - Clear Header 249, 248, 270
    - Edit Repetitions 265, 271, 273, 272, 230, 273
    - First Mkr Point 254, 256, 269
    - Gate Active Low High 269, 267
    - Header RMS 250
    - I/Q Mod Filter Manual Auto 253, 251, 264, 265
    - Last Mkr Point 254, 256
    - Marker 1 2 3 4 254, 260, 255, 256, 258, 259, 252
    - Name and Store 265
    - noise 261, 262
    - None 258, 259
    - Off 264
    - Patt Trig In 1 272
    - Reference Freq 263, 328, 269
    - Save Setup To Header 251, 265, 267, 274, 275, 255, 254, 256, 267, 270
    - Through 251, 253, 265, 269, 271
    - Waveform Runtime Scaling 264
  - Dual-Sine softkey 173, 177, 185
  - dwelt points 129
  - dwelt time 129
- ## E
- E8241A
    - 44A,51A,54A 349
  - E8247C,57C,67C 349
  - echo state 26
  - EDGE softkey 287, 289
  - Edit Repetitions softkey 265
  - Enter Secure Mode softkey 101
  - Erase All softkey 100
  - Erase and Overwrite All softkey 102
  - Erase and Sanitize All softkey 102
  - Erase softkey 100
  - ERROR

---

# Index

- 221 [53](#)
- Error Info softkey [91, 92](#)
- Event 1 Polarity Neg Pos [73, 75](#)
- Event 2 Polarity Neg Pos [73, 75](#)
- Execute Cal softkey [16, 18, 19, 20](#)
- Ext 50 Ohm softkey [233, 246](#)
- Ext 600 Ohm softkey [233, 246](#)
- Ext BBG Ref Freq softkey
  - See* custom subsystem keys
- Ext Data Clock Normal Symbol softkey
  - See* custom subsystem keys [217](#)
- Ext Delay Bits softkey [228](#)
- Ext Delay Off On softkey
  - custom subsystem [228](#)
  - Dmodulation subsystem [296](#)
  - dual ARB subsystem [273](#)
- Ext Delay Time softkey [272, 295](#)
- Ext Detector Coupling Factor softkey [147](#)
- Ext In 600 Ohm I Offset softkey [236](#)
- Ext In 600 Ohm Q Offset softkey [237](#)
- Ext Polarity Neg Pos softkey
  - custom subsystem [229](#)
  - Dmodulation subsystem [296](#)
  - dual ARB subsystem [230, 273](#)
- Ext Polarity Normal Inverted softkey
  - pulse modulation subsystem [190](#)
- Ext softkey
  - custom subsystem [214, 227](#)
  - Dmodulation subsystem [294](#)
  - dual ARB subsystem [271](#)
  - List/Sweep subsystem [133](#)
  - low frequency output subsystem [179](#)
  - trigger subsystem [106](#)
- Ext softkeys
  - Ext [163, 171, 160, 169, 182, 160, 169, 183, 195, 164, 187, 164, 173](#)
- extended numeric SCPI parameter [6](#)
- external filter [231](#)
- external frequency reference [285](#)
- external module start frequency [94](#)
- external module stop frequency [94](#)
- External Ref Bandwidth softkey [126](#)
- external reference oscillator [126](#)
- external trigger source
  - custom subsystem [227](#)
  - Dmodulation subsystem [294](#)
  - dual ARB subsystem [271](#)
- F**
- Fall Delay softkey
  - See* custom subsystem keys
- Fall Time softkey [210](#)
  - See* custom subsystem keys
- file
  - names [10, 44, 60](#)
  - retrieval [53](#)
  - systems [59](#)
  - types [59](#)
- file names [11](#)
- filename [11](#)
- Filter Alpha softkey
  - See* custom subsystem keys
  - See* Dmodulation subsystem keys
  - See* GPS subsystem keys
- Filter BbT softkey
  - See* custom subsystem keys
  - See* Dmodulation subsystem keys
  - See* GPS subsystem keys
- filter external [231](#)
- filters
  - digital modulation subsystem [231, 243](#)
  - Dmodulation subsystem [277, 280](#)
  - dual ARB subsystem [251, 253](#)
  - multitone subsystem [298, 300](#)
  - two tone subsystem [322, 324](#)
- FIR data [47](#)
- FIR softkey [40](#)
- firmware revision [29](#)
- First Mkr Point softkey [254, 256](#)
- FIX4 softkey [214, 215](#)
  - See* custom subsystem keys
  - See* GPS subsystem keys
- fixed frequency [119](#)
- fixed power [148](#)
- flat noise bandwidth [261](#)
- Flatness Off On softkey [112](#)
- flatness preset [111](#)
- FM softkeys
  - FM ΦM Normal High BW [182, 174, 175, 174, 168, 172, 171, 172, 170](#)
- forgiving listening and precise talking [5](#)
- free run [269](#)
- Free Run softkey
  - AM trigger source [163](#)
  - custom subsystem [225](#)
  - Dmodulation subsystem [292](#)
  - dual ARB subsystem [269](#)
  - FM trigger source [171](#)
  - list trigger source [133, 179](#)
  - phase modulation subsystem [186](#)
  - trigger subsystem [106](#)
- Freq Channels softkey [114, 116](#)
- Freq CW softkey [119](#)
- Freq Dev softkey
  - See* custom subsystem keys
  - See* Dmodulation subsystem keys
- Freq Separation softkey [321](#)
- Freq Spacing softkey [287, 306, 307](#)

- Freq Span softkey [122](#)
  - frequency
    - CW mode [124](#)
    - internal modulation [176](#)
    - list sweep points [130](#), [131](#)
    - mode [119](#)
    - reference [121](#)
    - start [122](#), [123](#)
  - frequency and amplitude correction pair [110](#)
  - Frequency hardkey [117](#), [124](#)
  - frequency modulation subsystem keys
    - Bus [171](#)
    - Dual-Sine [173](#)
    - Ext [171](#), [169](#), [173](#)
    - FM Dev [174](#), [175](#), [174](#), [168](#), [172](#), [173](#), [172](#), [171](#), [172](#), [170](#), [171](#)
    - Gaussian [172](#)
    - Incr Set [169](#), [173](#)
    - Negative [173](#)
    - Positive [173](#)
    - Ramp [173](#)
    - Sine [173](#)
    - Triangle [173](#), [171](#)
    - Uniform [172](#)
  - frequency multiplier [96](#)
  - frequency subsystem [113](#)
  - frequency subsystem keys
    - Adjust Phase [125](#)
    - External Ref Bandwidth [126](#)
    - Freq Center [113](#), [114](#), [116](#), [119](#), [118](#), [119](#), [117](#), [120](#), [121](#), [122](#), [123](#), [117](#), [124](#)
    - Internal Ref Bandwidth [126](#)
    - Phase Ref Set [125](#)
    - Ref Oscillator Source Auto Off On [127](#), [126](#)
    - Sweep Type [119](#)
  - FSK softkey [40](#)
  - Function Generator softkey [180](#)
  - function shape [173](#)
- G**
- gain [235](#), [237](#)
  - Gate Active Low High softkey
    - custom subsystem [226](#)
    - Dmodulation subsystem [293](#)
    - dual ARB subsystem [269](#)
  - gated [291](#)
  - Gated softkey
    - custom subsystem keys [224](#)
    - Dmodulation subsystem [291](#)
    - dual ARB subsystem [267](#)
  - gated trigger [267](#)
  - gated trigger mode
    - custom subsystem [224](#)
    - Dmodulation subsystem [291](#)
    - dual ARB subsystem [267](#)
  - gateway [22](#)
  - Gaussian [162](#), [184](#)
  - Gaussian softkey [172](#), [178](#)
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
    - See* GPS subsystem keys
  - generate sine [249](#)
  - Goto Row softkey [309](#)
  - GPiB Address softkey [21](#)
  - GPS Ref (f0) softkey [316](#), [319](#)
  - GPS Ref Clk Ext Int softkey [316](#), [319](#)
  - GPS subsystem
    - Data Mode Raw Enc TLM [312](#)
  - GPS subsystem keys
    - APCO 25 C4FM [312](#)
    - Data Mode Raw Enc TLM [311](#), [312](#)
    - Filter Alpha [313](#), [314](#), [311](#)
    - Gaussian [312](#), [316](#)
    - IQ Phase Normal Invert [315](#), [312](#)
    - Nyquist [312](#)
    - Optimize FIR For EVM ACP [314](#)
    - P Code Pwr [315](#), [311](#)
    - Ranging Code C/A P C/A+P [315](#), [317](#), [312](#)
    - Satellite ID [316](#)
    - UN3/4 GSM Gaussian [312](#), [311](#), [312](#)
  - Gray Coded QPSK softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
  - GSM softkey [287](#), [289](#)
  - GTLOCAL [21](#)
- H**
- header description [62](#)
  - header ID [63](#)
  - Help Mode Single Cont softkey [92](#)
  - hexadecimal values [13](#)
  - High Crest Mode Off On softkey [232](#)
  - hostname softkey [22](#)
- I**
- I offset external [236](#)
  - I Offset softkey [238](#), [330](#)
  - I/Q Adjustments Off On softkey [241](#), [331](#)
  - I/Q calibration [16](#), [17](#)
  - I/Q Calibration softkey [16](#)
  - I/Q calibration start stop [18](#)
  - I/Q clipping [248](#)
  - I/Q Gain Balance Source 1 softkey [237](#)
  - I/Q Mod Filter Manual Auto softkey [200](#), [244](#), [253](#), [280](#), [301](#), [324](#)
  - I/Q Off On softkey [247](#), [331](#)
  - I/Q Out Gain Balance softkey [235](#)

---

# Index

I/Q Output Atten softkey [236](#)  
I/Q Output Filter Manual Auto softkey [198](#), [231](#), [251](#), [276](#),  
[299](#), [322](#)  
I/Q Scaling softkey  
    *See* custom subsystem keys  
I/Q softkey [41](#)  
I/Q Timing Skew Path [240](#)  
I/Q timing Skew softkey [239](#)  
IDN command [92](#)  
IEEE 488.2 common command keys  
    Diagnostic Info [35](#)  
    RECALL Reg [36](#), [38](#)  
    Save Reg [37](#), [36](#)  
IEEE 488.2 common commands  
    CLS [34](#)  
    ESE [34](#), [35](#)  
    OPC [35](#)  
    PSC [36](#)  
    RST [36](#)  
    SAV [37](#), [38](#)  
    TRG [38](#)  
    WAI [38](#)  
Immediate softkey [264](#), [286](#)  
Incr Set hardkey [158](#), [167](#), [169](#), [191](#)  
    *See* phase modulation subsystem keys  
Initialize Phase Fixed Random softkey [308](#)  
Initialize Table softkey [289](#)  
Insert Row softkey [289](#)  
Insert Waveform softkey [265](#)  
Installed Board Info softkey [28](#)  
Int I/Q Skew Corrections RF BB Off softkey [245](#)  
Int softkeys  
    Int Doublet [194](#), [195](#), [194](#), [195](#), [194](#), [195](#), [232](#), [244](#), [194](#), [195](#)  
integer response data [8](#)  
interface module  
    *See* digital subsystem keys  
Internal Ref Bandwidth softkey [126](#)  
Internal softkeys  
    Internal 1 [187](#), [164](#), [173](#), [187](#), [180](#), [194](#), [195](#)  
Inverse Video Off On softkey [32](#)  
IP address [22](#)  
IP Address softkey [23](#)  
IQ Delay softkey [233](#)  
IQ Phase Normal Invert softkey [315](#), [318](#)  
IS-95 Mod softkey  
    *See* GPS subsystem keys  
IS-95 Mod w/EQ softkey  
    *See* GPS subsystem keys  
IS-95 OQPSK softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys  
IS-95 QPSK softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys

IS-95 softkey  
    *See* GPS subsystem keys  
IS-95 w/EQ softkey  
    *See* GPS subsystem keys

## L

LAN Config softkey [22](#)  
LAN, hostname [22](#)  
Language softkey [93](#), [97](#)  
Last Mkr Point softkey [254](#), [256](#)  
Leveling Mode softkey [146](#)  
LF Out softkeys  
    LF Out Amplitude [176](#), [180](#), [176](#), [177](#), [183](#), [179](#), [177](#), [176](#), [177](#),  
    [183](#)  
License Info softkey [28](#)  
list data [52](#)  
list frequency mode [119](#)  
list power mode [148](#)  
List softkey [41](#), [59](#)  
list sweep data [58](#)  
list/sweep subsystem [128](#)  
Load From Selected File softkey [57](#), [63](#), [110](#), [305](#)  
load list data [63](#)  
Load List From Step Sweep softkey [134](#)  
Load/Store softkey [288](#)  
local [21](#)  
Local hardkey  
    communication subsystem [21](#)  
lock and level [66](#), [67](#)  
low frequency output subsystem keys  
    Bus [179](#)  
    DC [177](#)  
    Ext [179](#)  
    Free Run [179](#), [180](#)  
    Gaussian [178](#)  
    Internal Monitor [180](#)  
    LF Out Amplitude [176](#), [180](#), [176](#), [177](#), [183](#), [179](#), [177](#), [176](#), [177](#),  
    [183](#)  
    Negative [178](#), [177](#)  
    Positive [178](#)  
    Ramp [177](#)  
    Sine [177](#)  
    Triangle [177](#), [179](#)  
    Uniform [178](#)  
Low Pass Filter below 2 GHz Off On [39](#)  
low-band filter subsystem keys  
    Low Pass Filter below 2 GHz Off On [39](#)

## M

Manual Freq softkey [118](#)  
Manual Mode Off On softkey [131](#), [137](#)  
Manual Point softkey [131](#)  
marker 1 2 3 4 [281](#), [301](#), [302](#)



- Marker 1 2 3 4 softkey [255, 256](#)
- Marker 1 Polarity Neg Pos softkey
  - Dmodulation subsystem [284](#)
  - dual ARB subsystem [260](#)
  - multitone subsystem [303](#)
  - two tone subsystem [327](#)
- Marker 1 softkey
  - Dmodulation subsystem [281, 282](#)
  - dual ARB subsystem [258, 259](#)
  - multitone subsystem [301, 302](#)
  - two tone subsystem [325, 326](#)
- Marker 1|2|3|4 Polarity Neg Pos softkey
  - AWGN ARB subsystem [203](#)
- Marker 1|2|3|4 softkey [200, 201](#)
- Marker 1|2|3|4 softkey [202](#)
- Marker 2 Polarity Neg Pos softkey
  - Dmodulation subsystem [284](#)
  - dual ARB subsystem [260](#)
  - multitone subsystem [303](#)
  - two tone subsystem [327](#)
- Marker 2 softkey
  - Dmodulation subsystem [281, 282](#)
  - dual ARB subsystem [258, 259](#)
  - multitone subsystem [301, 302](#)
  - two tone subsystem [325, 326](#)
- Marker 3 Polarity Neg Pos softkey
  - Dmodulation subsystem [284](#)
  - dual ARB subsystem [260](#)
  - multitone subsystem [303](#)
  - two tone subsystem [327](#)
- Marker 3 softkey
  - Dmodulation subsystem [281, 282](#)
  - dual ARB subsystem [258, 259](#)
  - multitone subsystem [301, 302](#)
  - two tone subsystem [325, 326](#)
- Marker 4 Polarity Neg Pos softkey
  - Dmodulation subsystem [284](#)
  - dual ARB subsystem [260](#)
  - multitone subsystem [303](#)
  - two tone subsystem [327](#)
- Marker 4 softkey
  - Dmodulation subsystem [281, 282](#)
  - dual ARB subsystem [258, 259](#)
  - multitone subsystem [301, 302](#)
  - two tone subsystem [325, 326](#)
- Marker Delta Off On softkey [140](#)
- Marker Freq softkey [140](#)
- Marker On Off softkey [141](#)
- marker polarity [303, 327](#)
- Marker softkey [254](#)
- marker subsystem [139](#)
- marker subsystem keys
  - Amplitude Markers Off On [139](#)
  - Delta Markers [140, 141](#)
- Marker Delta Off On [140, 141, 139](#)
- Turn Off Markers [139](#)
- Marker Value softkey [139](#)
- Markers [139, 140, 141, 254, 282](#)
- markers
  - ALC hold [201](#)
    - Dmodulation subsystem [281](#)
    - dual ARB subsystem [258](#)
    - multitone subsystem [301](#)
    - two tone subsystem [325](#)
  - alternate amplitude
    - AWGN subsystem [200](#)
  - AWGN ARB subsystem [202, 201](#)
  - clear all [255, 254](#)
  - marker polarity
    - AWGN subsystem [203](#)
    - Dmodulation subsystem [284](#)
    - dual ARB subsystem [260](#)
    - multitone subsystem [303](#)
    - two tone subsystem [327](#)
  - RF blanking/pulse [202](#)
    - Dmodulation subsystem [282](#)
    - dual ARB subsystem [259](#)
    - multitone subsystem [302](#)
    - two tone subsystem [326](#)
  - setting [256, 255](#)
- mass memory subsystem keys
  - Binary [59](#)
  - Delete All NVWFM Files [61, 348, 61](#)
  - List [59, 63](#)
  - State [59, 63](#)
  - User Flatness [59](#)
- Master softkey [135](#)
- mcarrier [289](#)
- MDMOD softkey [41](#)
- measurement units [108](#)
- memory [11](#)
- memory subsystem [46, 47, 49, 57](#)
- memory subsystem keys [51, 52](#)
  - Add Comment To Seq[n] Reg[nn] [58, 43, 57](#)
  - Binary [39](#)
  - Copy [44, 60](#)
  - Data PRAM [50, 55, 56, 54, 55, 56, 57, 61, 57, 40](#)
  - FIR [40](#)
  - I/Q catalog [41](#)
  - List [41, 57](#)
  - MDMOD [41, 42](#)
  - Oversample Ratio [47](#)
  - Rename File [58, 63](#)
  - SEQ [42, 43, 58](#)
  - User Flatness [43](#)
- Meter Address softkeys [24](#)
- Meter Channel A B softkey [24](#)
- Meter Timeout softkey [25](#)

---

# Index

Mod On/Off hardkey 66  
modulation 221  
modulation off on 66  
Modulator Atten (nnn dB) Manual Auto softkey 242  
Modulator Atten Manual Auto softkey 199, 241, 252, 279, 299, 300, 323  
Modulator I/Q Output Atten softkey 242  
module, digital signal interface  
    *See* digital subsystem keys  
move, files 58, 63  
MSGPS subsystem keys  
    GPS Ref (f0) 319  
    IQ Phase Normal Invert 318  
    Number of Satellites 320  
    Pause/Resume 318  
    Real-time MSGPS Off On 320, 319  
    Scenario 320  
MSK softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys  
MSUS 11, 59  
MTONE softkey 42  
Multicarrier Off On softkey 287  
multicarrier setup 287  
multiplier 95  
multitone markers, *See* markers  
Multitone Off On softkey 310  
multitone subsystem keys  
    40.000 MHz 298, 300, 324  
    ARB Reference Ext Int 304, 328  
    Clear Header 298  
    Freq Spacing 306, 307  
    Goto Row 309  
    I/Q Mod Filter Manual Auto 301, 299, 308  
    Load From Selected File 305  
    Marker 1 301, 302, 303, 301, 302, 303, 301, 302, 303, 301, 302, 303, 299, 300, 310  
    None 301, 302, 306, 308  
    Random Seed Fixed Random 309, 304  
    Save Setup To Header 298, 306  
    Through 298, 300, 322, 324, 306, 309  
mV softkey 108  
mVemf softkey 108

## N

N5102A  
    bit order 337  
    clock phase 334, 335, 336  
    data type 344, 337  
    frame polarity 340  
    I gain 338  
    i offset 339, 340  
    logic type 345, 344

    N5102A off on 346  
    negate I 338, 341, 339  
    pass through 346, 345  
    Q gain 341, 342  
    reference frequency 335, 342  
    scaling 342, 343, 339  
    word alignment 336, 343  
NADC softkey 287, 289  
Name and Store softkey 265  
Negative softkey 162, 173, 178, 184  
noise 261, 262  
    noise bandwidth 261  
Noise Seed Fixed Random softkey 205  
Noise softkey 173, 177, 185, 186  
noise state on 261  
None softkey 100, 200, 201, 202, 223, 258, 259, 281, 282, 301, 302, 325, 326  
non-volatile memory 11  
Normal Inverted Polarity 190  
Number of Satellites softkey 320  
Number Of Tones softkey 306, 308  
numeric boolean response data 8  
numeric SCPI parameter 6  
numeric, extended SCPI parameter 6  
Nyquist softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys  
    *See* GPS subsystem keys

## O

octal values 13  
OEM  
    frequency band 95  
    multiplier 96  
    on off, select 94  
    start 94  
Off softkey 233, 246, 264, 286  
offset frequency 120  
offset, common mode 234  
offset, differential Q 235  
offset, ext I/Q signal 234  
On softkey 264, 286  
Optimize FIR For EVM ACP softkey  
    *See* custom subsystem keys  
    *See* Dmodulation subsystem keys  
    *See* GPS subsystem keys  
Optimize for (nnn sps) Bandwidth softkey 243  
Optimize S/N Ratio  
    softkey 149  
options  
    007, marker subsystem 139, 330  
    403  
        AWGN real-time subsystem 205, 198

- GPS subsystem [311](#)
  - MSGPS subsystem [318](#)
  - 601 or 602
    - all subsystem [197, 321](#)
    - custom subsystem [206](#)
    - digital modulation subsystem [231](#)
    - Dmodulation subsystem [276](#)
    - dual ARB subsystem [247](#)
    - multitone subsystem [298](#)
  - Options Info softkey [29](#)
  - OQPSK softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
  - oscillator
    - bandwidth [126](#)
    - reference [126](#)
    - source [127](#)
  - Output Blanking Off On Auto softkey [65](#)
  - output settled, polarity [66](#)
  - output settled, query [66](#)
  - output settled, retrace [67](#)
  - output settled, RF off [67](#)
  - output subsystem
    - output settled [66](#)
    - polarity, output settled [66](#)
    - retrace, output settled [67](#)
    - RF off, output settled [67](#)
  - output subsystem keys
    - Mod On/Off [66](#)
    - Output Blanking Off On Auto [65](#)
    - RF On/Off [68](#)
  - Oversample Ratio softkey [47](#)
  - Overwrite softkey [100](#)
- P**
- P Code Pwr softkey [315](#)
  - parameter types. *See* SCPI commands parameter types
  - paths, SCPI command tree [5](#)
  - Patt Trig In 1 softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
    - See* dual ARB subsystem keys
  - Patt Trig In 2 [272](#)
  - Patt Trig In 2 softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
    - See* dual ARB subsystem keys
  - Pause/Resume softkey [318](#)
  - PDC softkey [287, 289](#)
  - persistent
    - power on states [98](#)
  - phase adjustment [125](#)
  - Phase Dev softkey
    - See* custom subsystem keys
  - phase modulation subsystem keys
    - $\Phi$ M Sweep Time [186, 182](#)
    - $\Phi$ M Dev [188](#)
    - $\Phi$ M Dev Couple Off On [189](#)
    - $\Phi$ M Off On [187](#)
    - $\Phi$ M Path 1 2 [181](#)
    - $\Phi$ M Tone 2 Ampl Percent of Peak [185](#)
    - $\Phi$ M Tone 2 Rate [183](#)
    - Bus [186](#)
    - Dual-Sine [185](#)
    - Ext Coupling DC AC [182, 183, 187](#)
    - Free Run [186](#)
    - Incr Set [181, 189, 187](#)
    - Noise [185, 186](#)
    - Ramp [185, 186](#)
    - Sine [185, 186, 185, 186, 185](#)
    - Triangle [185, 186](#)
  - Phase Polarity Normal Invert softkey [221](#)
  - Phase Ref Set softkey [125](#)
  - PHS softkey [287, 289](#)
  - PN11 softkey
    - See* custom subsystem keys
  - PN15 softkey
    - See* custom subsystem keys
    - See* GPS subsystem keys
  - PN20 softkey
    - See* custom subsystem keys
  - PN23 softkey
    - See* custom subsystem keys
  - PN9 Mode Preset softkey [98](#)
  - PN9 softkey
    - See* custom subsystem keys
    - See* GPS subsystem keys
  - points
    - dwell [129](#)
    - selection [131](#)
  - polarity
    - burst gate [69, 70](#)
    - data clock input [69, 71, 72, 73, 69, 71, 72, 74, 244](#)
    - event 1 2 3 4 [73, 75](#)
    - I/Q [232](#)
  - markers
    - AWGN ARB subsystem [203](#)
    - Dmodulation subsystem [284](#)
    - dual ARB subsystem [260](#)
    - multitone subsystem [303](#)
    - two tone subsystem [327](#)
  - symbol sync input [70, 71, 73, 75](#)
  - triggers
    - custom subsystem [226, 229](#)
    - Dmodulation subsystem [293, 296](#)
    - dual ARB subsystem [269, 273](#)
  - Positive softkey [162, 173, 178, 184](#)

---

# Index

## power

- list sweep query [132](#)
- start [152](#)
- units [108](#)

## power meter

- address [24](#)
- channel B [24](#)
- timeout
  - GPIB [25](#)

## Power Meter softkey [25](#)

## Power On Last Preset softkey [96](#)

## power on states [98](#)

## Power Search Manual Auto softkey [143](#), [146](#)

## Power Search Reference Fixed Mod softkey [144](#), [145](#)

## power subsystem [142](#)

## power subsystem keys [150](#)

- ALC BW [142](#), [147](#), [153](#), [151](#), [152](#), [153](#), [154](#), [148](#)
- Do Power Search [143](#), [144](#), [145](#), [146](#)
- Ext Detector Coupling Factor [147](#)
- Leveling Mode [146](#)
- Power Search Manual Auto [143](#), [144](#), [145](#), [146](#)
- Set ALC Level [143](#), [147](#)

## power-on [28](#)

## PRAM

- data pattern [215](#), [50](#)
- list [51](#)

## PRAM DATA BLOCK [52](#)

## PRAM LIST [52](#)

## PRAM? [51](#)

## precise talking and forgiving listening [5](#)

## predefined setups, custom subsystem [223](#)

## preset [80](#)

## Preset hardkey [97](#)

## Preset List softkey [111](#), [134](#)

## Preset Normal User softkey [99](#)

## preset states [98](#)

## protection state [150](#)

## Pulse Frequency [191](#)

## pulse modulation subsystem [190](#), [191](#)

## pulse modulation subsystem keys [193](#)

- Delay Step [191](#)
- Ext Pulse [195](#)
- Int Doublet [194](#), [195](#), [194](#), [195](#), [194](#), [195](#), [194](#), [195](#), [194](#), [195](#)
- Pulse Delay [190](#), [195](#), [192](#), [191](#), [193](#)

## Pulse Period Increment [193](#)

## Pulse/RF blanking [259](#)

## pulse/RF blanking [202](#)

## pulse/rf blanking [202](#)

## pulse/RF blanking markers

- Dmodulation subsystem [282](#)
- dual ARB subsystem [259](#)
- multitone subsystem [302](#)
- two tone subsystem [326](#)

## PWT softkey [287](#), [289](#)

## Q

## Q external offset [237](#)

## Q Offset softkey [238](#), [330](#)

## QPSK softkey

- See* custom subsystem keys
- See* Dmodulation subsystem keys

## Quadrature Skew softkey [239](#), [331](#)

## query

- frequency points [131](#)
- power points [132](#)

## Query, IDN? [92](#)

## quotes, SCPI command use of [12](#)

## R

## ramp positive/negative [173](#)

## Ramp softkey [173](#), [177](#), [185](#), [186](#)

## ramp sweep [137](#)

- range [122](#)
- selecting [136](#), [122](#)
- time [138](#)

## ramp, low frequency [178](#)

## Random Seed Fixed Random softkey [309](#)

## Ranging Code C/A P C/A+P softkey [315](#)

## ratio, source [246](#)

## real response data [8](#)

## Real-time AWGN Off On softkey [205](#)

## real-time AWGN subsystem keys

- Bandwidth [205](#)
- Real-time AWGN Off On [205](#)

## real-time custom triggering, *See* triggers

## Real-time GPS Off On softkey [317](#)

## Real-time MSGPS Off On softkey [320](#)

## real-time noise [261](#)

## RECALL Reg softkey [36](#)

## recall state files [58](#)

## Rectangle softkey

- See* custom subsystem keys
- See* Dmodulation subsystem keys
- See* GPS subsystem keys

## rectangular waveguide [95](#)

## Ref Oscillator Source Auto Off On softkey [127](#)

## Reference Freq softkey

- See* AWGN subsystem keys
- See* Dmodulation subsystem keys
- See* dual ARB subsystem keys
- See* multitone subsystem keys

## reference oscillator bandwidth [126](#)

## reference oscillator internal [126](#)

## remote [33](#)

## Rename File softkey [58](#), [63](#)

## reset & run [269](#), [292](#)

## Reset & Run softkey

- Dmodulation subsystem [292](#)

- dual ARB subsystem [269](#)
  - Reset RS-232 softkey [26](#)
  - response data types. *See* SCPI commands response types
  - Restart softkey [319](#)
  - Restore Factory Defaults softkey [126](#)
  - Restore Sys Defaults softkey [98](#)
  - Retrace Off On softkey [132](#)
  - retrace, sweeps [132](#)
  - retrigger, single mode [264](#), [286](#)
  - Revert to Default Cal Settings softkey [17](#), [19](#)
  - revision number, firmware [29](#)
  - rf blanking [259](#)
  - RF blanking/pulse markers
    - Dmodulation subsystem [282](#)
    - dual ARB subsystem [259](#)
    - multitone subsystem [302](#)
    - two tone subsystem [326](#)
  - RF On/Off hardkey [68](#)
  - Rise Delay softkey [211](#)
    - See* custom subsystem keys
  - Rise Time softkey [212](#)
    - See* custom subsystem keys
  - RMS header info [250](#)
  - Root Nyquist softkey
    - See* custom subsystem keys
    - See* Dmodulation subsystem keys
    - See* GPS subsystem keys
  - rotate markers [255](#)
  - route subsystem keys
    - Burst Gate In Polarity Neg Pos [69](#), [70](#)
    - Data Clock Out Neg Pos [72](#), [69](#), [71](#), [73](#), [72](#), [74](#), [69](#), [71](#), [74](#)
    - Event 1 Polarity Neg Pos [73](#), [75](#), [73](#), [75](#)
    - Symbol Sync Out Polarity Neg Pos [73](#), [75](#), [70](#), [71](#)
  - RS-232 Baud Rate softkey [26](#)
  - RS-232 ECHO Off On softkeys [26](#)
  - RS-232 reset [26](#)
  - RS-232 Timeout softkeys [27](#)
  - Run Complete Self Test softkey [38](#)
  - runtime scaling [264](#)
- S**
- Sanitize softkey [100](#)
  - Satellite ID softkey [316](#)
  - save flatness data [111](#)
  - Save Reg softkey [37](#)
  - Save Seq[n] Reg[nn] softkey [37](#)
  - Save Setup To Header softkey [199](#), [251](#), [279](#), [298](#), [322](#)
  - save state files [58](#)
  - Save User Preset softkey [99](#)
  - Scale Waveform Data softkey [265](#)
  - scaling
    - during playback [264](#)
    - waveform files [265](#)
  - Scaling softkey [265](#)
  - Scenario softkey [320](#)
  - SCPI
    - backward compatible
      - \*IDN? output [347](#)
      - 8340B/41B [350](#), [363](#), [379](#), [387](#), [399](#), [350](#)
    - [2](#), [13](#)
    - command tree [4](#), [5](#), [4](#), [10](#), [2](#)
      - 8257D/67D/E8663D [348](#)
    - errors [91](#), [92](#)
    - hexadecimal [13](#)
    - MSUS variable [11](#)
    - octal [13](#), [1](#)
    - parameter and response types [5](#)
      - boolean [7](#)
      - discrete [7](#)
      - extended numeric [6](#)
      - numeric [6](#)
      - string [8](#)
    - [5](#), [9](#)
    - quote usage [12](#)
    - response data types
      - discrete [8](#)
      - integer [8](#)
      - numeric boolean [8](#)
      - real [8](#)
      - string [9](#)
    - [5](#)
    - syntax [2](#)
    - version, system subsystem [104](#)
  - SCPI command subsystems
    - all [197](#), [321](#), [330](#), [157](#)
    - AWGN [198](#), [205](#)
    - calibration [16](#), [21](#), [110](#), [206](#)
    - diagnostic [28](#), [231](#), [333](#), [30](#)
    - Dmodulation [276](#)
    - dual ARB [247](#)
    - frequency [113](#), [168](#)
    - GPS subsystem [311](#)
    - IEEE 488.2 common commands [34](#)
    - list/sweep [128](#), [176](#), [39](#)
    - marker [139](#), [59](#), [39](#)
    - MSGPS subsystem [318](#)
    - multitone [298](#)
    - output [65](#)
    - phase modulation [181](#), [142](#), [190](#)
    - route [69](#)
    - status [76](#), [90](#)
    - trigger [105](#)
    - Tsweep [155](#)
  - screen blanking [33](#)
  - screen capture [32](#)
  - Screen Saver Delay

---

# Index

- 1 hr softkey [102](#)
- Screen Saver Mode softkeys [103](#)
- Screen Saver Off On softkeys [103](#)
- secure wave directory [53](#)
- security functions
  - erase [100](#)
  - none [100](#)
  - overwrite [100](#), [102](#)
  - sanitize [100](#), [102](#), [99](#), [101](#)
- segment advance [267](#)
  - softkey [267](#)
  - trigger mode [267](#), [291](#), [270](#)
- Select File softkey [287](#)
- Select Scenario softkey [320](#)
- Select Seq softkey [36](#)
- Select Waveform softkey [274](#), [275](#)
- SEQ softkey [42](#)
- sequence files [11](#)
- sequence, creating [265](#)
- Set ALC Level softkey [143](#)
- Set Atten softkey [147](#)
- Set Marker Off All Points softkey [255](#)
- Set Marker Off Range Of Points softkey [254](#)
- Set Marker On Range Of Points softkey [256](#)
- setting markers [256](#)
- SHAPE softkey [42](#)
- shift markers [255](#)
- signal to noise ratio
  - setting [149](#)
- Sine softkey [173](#), [213](#)
  - See* low frequency output subsystem keys
  - See* phase modulation subsystem keys
- single [291](#)
  - segment advance [270](#)
  - trigger mode
    - custom subsystem [224](#)
    - Dmodulation subsystem [291](#)
    - dual ARB subsystem [267](#)
- [264](#), [286](#)
- Single softkey
  - custom subsystem [224](#)
  - Dmodulation subsystem [291](#)
  - dual ARB subsystem [267](#), [270](#)
- Single Sweep softkey [105](#), [155](#)
- skew [239](#), [240](#)
- skew, I/Q
  - adjustment [239](#)
  - path [245](#)
  - state [245](#)
- Slave softkey [135](#)
- software options [28](#)
- source
  - bbg1 [233](#)
  - external [233](#)
  - internal [233](#)
  - sum [233](#), [246](#)
- source I/Q modulator [246](#)
- source trigger
  - custom subsystem [227](#)
  - Dmodulation subsystem [294](#)
  - dual ARB subsystem [271](#)
- Span Type User Full softkey [146](#)
- Square softkey [173](#), [177](#), [185](#), [186](#)
- start frequency [122](#)
- Start Frequency softkey [18](#), [20](#), [145](#)
- State softkey [43](#), [59](#)
- Status Byte Register commands
  - IDN? [35](#)
  - RCL [36](#)
- status register commands [76–89](#)
- step and list frequencies [119](#)
- step and list power [148](#)
- Step Dwell softkey [136](#)
- step sweep
  - selecting [131](#), [136](#)
- stop frequency [123](#)
- Stop Frequency softkey [18](#), [20](#), [145](#)
- Store Custom Dig Mod State softkey [290](#)
- store list data [63](#)
- Store To File softkey [58](#), [63](#), [111](#), [306](#)
- string response data [9](#)
- string SCPI parameter [8](#)
- strings, quote usage [12](#)
- Subnet Mask softkey [23](#)
- subsystems
  - correction [110](#)
  - frequency [113](#)
  - list/sweep [128](#)
  - marker [139](#)
  - power [142](#)
  - Tsweep [155](#)
- Summing Ratio (SRC1/SRC2) x.xx dB softkey [246](#)
- sweep
  - abort [155](#)
  - commands [128–138](#)
  - Control softkey [134](#)
  - Direction Down Up softkey [128](#)
  - rate [171](#)
  - Retrace Off On softkey [132](#)
  - Time Manual Auto softkey [138](#), [133](#), [136](#), [148](#)
- Sweep Repeat Single Cont softkey [105](#)
- Swept-Sine softkey [173](#), [177](#), [185](#)
- Symbol Out Polarity Neg Pos softkey [73](#)
- Symbol Rate softkey [290](#)
- Symbol Sync Out Polarity Neg Pos softkey [75](#)
- Symbol Sync Polarity Neg Pos softkey [70](#), [71](#)
- system
  - capability [90](#)

date [91](#)  
   preset [97](#)  
 system commands [90–104](#)  
 system subsystem keys  
   8648A/B/C/D [93, 97, 93, 97, 93, 97, 93, 97, 93, 97](#)  
   Activate Secure Display [99, 90](#)  
   Diagnostic Info [92](#)  
   Enter Secure Mode [101](#)  
   erase [100](#)  
   Erase All [100, 102, 91, 92](#)  
   Help Mode Single Cont [92](#)  
   none [100](#)  
   overwrite [100](#)  
   PN9 Mode Preset [98, 96, 97, 99](#)  
   Restore Sys Defaults [98](#)  
   sanitize [100](#)  
   Save User Preset [99, 93, 97](#)  
     1 hr [102](#)  
     [103](#)  
   Time/Date [91, 103](#)  
   View Next Error Message [91, 92](#)

## T

table setup, multitone [306](#)  
 TETRA softkey [287, 289](#)  
 through [298, 300, 322, 324](#)  
 Through softkey [198, 200, 231, 243, 251, 253, 276, 280, 298, 300, 322, 324](#)  
 time, dwell [129](#)  
 Time/Date softkey [91, 103](#)  
 timeout RS-232 [27](#)  
 Toggle Marker 1 2 3 4 softkey [265](#)  
 toggle state [306](#)  
 Toggle State softkey [306, 309](#)  
 Triangle softkey [173, 177, 185, 186](#)  
 trigger [294](#)  
   segment advance [267](#)  
 Trigger & Run softkey  
   custom subsystem [225](#)  
   Dmodulation subsystem [292](#)  
   dual ARB subsystem [269](#)  
 trigger commands [105–107](#)  
 trigger custom  
   free [225](#)  
   single [224](#)  
 Trigger In Polarity Neg Pos softkey [106](#)  
 Trigger Key softkey  
   Dmodulation subsystem [294](#)  
   dual ARB subsystem keys [271](#)  
   frequency modulation subsystem [171](#)  
   list/sweep subsystem [133, 179](#)  
   phase modulation subsystem [186](#)  
   trigger subsystem [106](#)  
   trigger key trigger source  
     custom subsystem [227](#)  
     Dmodulation subsystem [294](#)  
     dual ARB subsystem [271](#)  
   Trigger Out Polarity Neg Pos softkey [106](#)  
   trigger source, list sweep [133](#)  
   trigger subsystem keys  
     Bus [106, 163](#)  
     Ext [106, 163](#)  
     Free Run [106, 163](#)  
     Single Sweep [105](#)  
     Trigger In Polarity Neg Pos [106](#)  
   trigger sweep  
     bus [163](#)  
   trigger sweep  
     external [163](#)  
     immediate [163](#)  
     key [163](#)  
 triggers  
   connector selection  
     custom subsystem [229](#)  
     Dmodulation subsystem [295](#)  
     dual ARB subsystem [272](#)  
   delay  
     custom subsystem [228](#)  
     Dmodulation subsystem [295, 296](#)  
     dual ARB subsystem [272, 273](#)  
   mode selection  
     custom subsystem [224](#)  
     Dmodulation subsystem [291](#)  
     dual ARB subsystem [267](#)  
   polarity selection  
     cont & single mode, custom [229, 296, 273](#)  
     gate mode, custom [226, 293, 269](#)  
   response selection  
     continuous mode, custom [225, 292, 269](#)  
     segment advance mode, dual ARB [270, 286, 264](#)  
   source selection  
     custom subsystem [227](#)  
     Dmodulation subsystem [294](#)  
     dual ARB subsystem [271](#)  
 Tsweep subsystem [155](#)  
 Turn Off Markers softkey [139](#)  
 two tone markers, *See* markers  
 two tone Off On softkey [329](#)  
 two tone subsystem keys  
   40.000 MHz [322](#)  
   Alignment Left Cent Right [321](#)  
   Clear Header [322](#)  
   Freq Separation [321](#)  
   I/Q Mod Filter Manual Auto [324, 322](#)  
   Marker 1 [325, 326, 327, 325, 326, 327, 325, 326, 327, 325, 326,](#)  
     [327, 323](#)  
   None [325, 326](#)

---

# Index

Save Setup To Header [322](#)  
two tone Off On [329](#)

## U

UN3/4 GSM Gaussian softkey  
*See* custom subsystem keys  
*See* Dmodulation subsystem keys  
*See* GPS subsystem keys  
Uniform softkey [162](#), [172](#), [178](#)  
unit subsystem keys  
  dBuV [108](#)  
  mV [108](#)  
  units [108](#)  
units [30](#), [108](#)  
unprotected  
  memory subsystem [53](#)  
unspecified RMS [250](#)  
Update in Remote Off On softkey [33](#)  
uploading files [53](#)  
User File softkey [213](#)  
  *See* custom subsystem keys  
  *See* GPS subsystem keys  
User FIR softkey  
  *See* custom subsystem keys  
  *See* Dmodulation subsystem keys  
  *See* GPS subsystem keys  
user flatness corrections, state [112](#)  
User Flatness softkey [43](#), [59](#)  
user flatness, delete files [57](#)  
User FSK softkey  
  *See* custom subsystem keys  
  *See* Dmodulation subsystem keys  
User I/Q softkey  
  *See* custom subsystem keys  
  *See* Dmodulation subsystem keys  
uV softkey [108](#)  
uVemf softkey [108](#)

## V

VCO Clock Ext Int softkey [230](#), [274](#)  
View Next Error Message softkey [91](#), [92](#)  
volatile memory [11](#)

## W

waveform  
  multitone [298](#)  
  sequence, dual ARB [265](#)  
waveform clipping [248](#)  
Waveform Length softkey [203](#)  
Waveform Runtime Scaling softkey [264](#)  
waveform scaling  
  during playback [264](#)  
  files [265](#)

waveform shape [173](#)  
waveguide [95](#)  
WB IQ Calibration [19](#)  
WB IQ calibration full [20](#)  
WB IQ calibration start [20](#)  
WB IQ calibration stop [20](#)  
Wide Band IQ Calibration [18](#)  
wideband digital modulation subsystem keys  
  I Offset [330](#), [331](#)  
  Q Offset [330](#), [331](#)  
window state [33](#)  
WR bands [95](#)