Where to Find the Latest Information

Documentation is updated periodically. For the latest information about these products, including instrument software upgrades, application information, and product information, browse to one of the following URLs, according to the name of your product:

http://www.keysight.com/find/fieldfox

To receive the latest updates by email, subscribe to Keysight Email Updates at the following URL:

http://www.keysight.com/find/MyKeysight

Information on preventing instrument damage can be found at:

www.keysight.com/find/PreventingInstrumentRepair

Is your product software up-to-date?

Periodically, Keysight releases software updates to fix known defects and incorporate product enhancements. To search for software updates for your product, go to the Keysight Technical Support website at:

http://www.keysight.com/find/fieldfoxsupport
# Table of Contents

Home - List of Commands by Mode .............................................................................................................. 1
See Also ......................................................................................................................................................... 1
SCPI Concepts and Tips .................................................................................................................................. 3
FieldFox Programming Tips .......................................................................................................................... 3
Correction Methods Explained.................................................................................................................... 4
Cal Methods .................................................................................................................................................. 5
Summary ..................................................................................................................................................... 6
How the FieldFox Error Queue Works ........................................................................................................ 7
Instrument Console (IC) ................................................................................................................................ 8
The Rules and Syntax of SCPI ........................................................................................................................ 9
SCPI Errors ................................................................................................................................................... 13
Examples ..................................................................................................................................................... 21
FieldFox Programming Examples ................................................................................................................ 21
NA Mode Setup ........................................................................................................................................... 21
VVM Mode Example .................................................................................................................................... 23
VVS Example ............................................................................................................................................. 24
Calibration Examples ..................................................................................................................................... 25
QuickCals .................................................................................................................................................... 28
ECal Calibration .......................................................................................................................................... 30
FOPS Measurement ....................................................................................................................................... 31
Guided Calibration ....................................................................................................................................... 32
Marker Example ......................................................................................................................................... 33
Memory Command Examples ....................................................................................................................... 34
C# Example Program ................................................................................................................................... 36
Read Block Data using Csharp ................................................................................................................... 43
Transfer Image to PC ................................................................................................................................... 44
MATLAB ....................................................................................................................................................... 45
MATLAB - Binary Block File Transfer Via VISA ......................................................................................... 45
MATLAB - Binary Block File Transfer Via LAN as Socket at Port 5025 ...................................................... 48
Python Example .......................................................................................................................................... 50
VEE ............................................................................................................................................................... 55
Get Formatted Data as Real 32 Binary Block............................................................................................ 55
Get Formatted Data as ASCII Output ......................................................................................................... 56
Trigger Synch Sweep Complete .................................................................................................................. 56
SpecAn Get Data as ASCII Output ............................................................................................................. 57
Commands by Mode ..................................................................................................................................... 59
CAT Mode Commands ................................................................................................................................... 59
NA Mode Commands .................................................................................................................................... 63
SA Mode Commands .................................................................................................................................... 70
CALCulate:MARKer<n>:FUNCTION <char> ........................................................................................... 141
CALCulate:MARKer:FUNCTION:BAND:SPAN <num> ........................................................................... 142
CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO <bool> ................................................................. 142
CALCulate:MARKer:FUNCTION:INTerval:SPAN <num> ....................................................................... 143
CALCulate:MARKer:FUNCTION:INTerval:SPAN:AUTO <bool> ............................................................. 143
CALCulate[:SELECTed]:MARKer<n>:FUNCTION:PLEFt .......................................................................... 144
CALCulate[:SELECTed]:MARKer<n>:FUNCTION:PRIGHT ................................................................. 144
CALCulate:MARKer<n>:NOISE[:STATE] <bool> - Superseded ........................................................... 144
CALCulate[:SELECTed]:MARKer<n>:SET:CENTer ............................................................................... 145
CALCulate:MARKer<n>:NOISE[:STATE] <bool> - Superseded ........................................................... 144
CALCulate:MEASURE:DATA? ................................................................................................................ 147
CALCulate:MEASURE:QAMPLitude? ...................................................................................................... 148
:CALCulate:PARameter:COUNT <n> ..................................................................................................... 148
:CALCulate:PARameter<tr>:DEFine <char> .......................................................................................... 149
:CALCulate:PARameter<n>:SELECT ...................................................................................................... 151
CALCulate:RELative[:MAGNitude]:AUTO <bool> ................................................................................. 151
:CALCulate:SPECtrum:MARKer:AOFF ................................................................................................ 152
:CALCulate:SPECtrum:MARKer:CPSearch[:STATE] <bool> ................................................................. 152
:CALCulate:SPECtrum:MARKer:DFixed <bool> ................................................................................... 152
:CALCulate:SPECtrum:MARKer:FUNCTION:PECursion <num>.......................................................... 153
:CALCulate:SPECtrum:MARKer:FUNCTION:PRIGHT .......................................................................... 153
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:BAND:SPAN <freq> ......................................................... 155
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:BAND:SPAN:AUTO ..................................................... 155
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:MAXimum ............................................................... 156
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:MINimum ............................................................... 156
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:PLEFt ........................................................................ 156
:CALCulate:SPECtrum:MARKer<n>:FUNCTION:PRIGHT ..................................................................... 157
:CALCulate:SPECtrum:MARKer<n>:PHASe? ....................................................................................... 157
:CALCulate:SPECtrum:MARKer<n>:SET:CENTer .............................................................................. 158
:CALCulate:SPECtrum:MARKer<n>:SET:RLEVEL ............................................................................. 158
:CALCulate:SPECtrum:MARKer<n>:TRACE <int> ................................................................................ 159
:CALCulate:SPECtrum:MARKer<n>:X <num> ..................................................................................... 160
:CALCulate:SPECtrum:MARKer<n>:Y? ............................................................................................... 160
:CALCulate:SPECtrum:MARKer<n>[:STATE] <char> ........................................................................... 160
:CALCulate:WAVEform:MARKer:AOFF .............................................................................................. 161
:CALCulate:WAVEform:MARKer:COUPLE:X <num> ......................................................................... 162
:CALCulate:WAVEform:MARKer:COUPLE[:STATE] <bool> ................................................................. 162
:CALCulate:WAVEform:MARKer:CPSearch[:STATE] <bool> ................................................................. 163
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CALCulate:WAVeform:MARKer:FUNtion:PEXCursion &lt;num&gt;</td>
<td>164</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer:FUNtion:PTHReshold &lt;num&gt;</td>
<td>164</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:DATA &lt;char&gt;</td>
<td>165</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:FUNtion &lt;char&gt;</td>
<td>165</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:FUNtion:INTerval:SPAN &lt;num&gt;</td>
<td>166</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:FUNtion:MAXimum</td>
<td>167</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:MARKer:FUNtion:Mn:MIN</td>
<td>167</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:FUNtion:PNEXT</td>
<td>168</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:TRACe &lt;int&gt;</td>
<td>169</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:X</td>
<td>169</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:Y?</td>
<td>170</td>
</tr>
<tr>
<td>:CALCulate:WAVeform:MARKer&lt;n&gt;:[STATe]</td>
<td>170</td>
</tr>
<tr>
<td>:DISPlay:TABLE:MARKer:SPECTrum &lt;bool&gt;</td>
<td>171</td>
</tr>
<tr>
<td>:DISPlay:TABLE:MARKer:WAVeform &lt;bool&gt;</td>
<td>172</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:AMPLitude:MARKer:DELTa:STATe &lt;bool&gt;</td>
<td>172</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:AMPLitude:MARKer:STATe &lt;bool&gt;</td>
<td>173</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:AMPLitude:MARKer:Y1 &lt;num&gt;</td>
<td>173</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:AMPLitude:MARKer:Y2 &lt;num&gt;</td>
<td>174</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:CONVersion:FUNtion &lt;char&gt;</td>
<td>174</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:CORRection:EDELay:TIME &lt;num&gt;</td>
<td>175</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:CORRection:OFFSet:PHASe &lt;num&gt;</td>
<td>175</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:DATA:FDATa &lt;data&gt;</td>
<td>176</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:DATA:FMEM &lt;data&gt;</td>
<td>176</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:DATA:SDATA &lt;data&gt;</td>
<td>177</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:DATA:SMEM &lt;data&gt;</td>
<td>177</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:CENTer &lt;num&gt;</td>
<td>177</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:SHAPE &lt;char&gt;</td>
<td>178</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:SPAN &lt;num&gt;</td>
<td>178</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:START &lt;num&gt;</td>
<td>179</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:STATe &lt;bool&gt;</td>
<td>179</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:STOP &lt;num&gt;</td>
<td>180</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FILTer:[GATE]:TIME:[TYPE] &lt;char&gt;</td>
<td>180</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FMEM:DATA?</td>
<td>181</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:FORMat &lt;char&gt;</td>
<td>181</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:GAIN:DATA??</td>
<td>182</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:BEEP &lt;bool&gt; - Superseded</td>
<td>182</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:DATA &lt;data&gt; - Superseded</td>
<td>183</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:LLData &lt;data&gt;</td>
<td>183</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:SOUNd &lt;char&gt;</td>
<td>185</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:STATe</td>
<td>185</td>
</tr>
<tr>
<td>:CALCulate:[SELelected]:LIMit:WARN &lt;bool&gt;</td>
<td>186</td>
</tr>
</tbody>
</table>
CALCulate[:SELected]:MARKer<n>:ACTivate ................................................................. 186
CALCulate[:SELected]:MARKer:AOFF ............................................................................. 187
CALCulate[:SELected]:MARKer:BWIDth:DATA? ............................................................... 187
CALCulate[:SELected]:MARKer:BWIDth[:STATE] <bool> .................................................. 188
CALCulate[:SELected]:MARKer:COUPled <bool> ............................................................. 188
CALCulate[:SELected]:MARKer:FORMat <char> .............................................................. 189
CALCulate[:SELected]:MARKer:FUNCtion:BWIDth:THReshold <value> ......................... 189
CALCulate[:SELected]:MARKer<n>:FUNCtion:FALLtime <num> ....................................... 190
CALCulate[:SELected]:MARKer<n>:FUNCtion:MAXimum ............................................... 190
CALCulate[:SELected]:MARKer<n>:FUNCtion:MINimum ................................................. 191
CALCulate[:SELected]:MARKer:FUNCtion:PEXCursion <num> ....................................... 191
CALCulate[:SELected]:MARKer<n>:FUNCtion:PLEFt .......................................................... 192
CALCulate[:SELected]:MARKer<n>:FUNCtion:PNEXt ........................................................ 192
CALCulate[:SELected]:MARKer:FUNCtion:PRIght ............................................................ 193
CALCulate[:SELected]:MARKer:FUNCtion:PTHReshold <num> ....................................... 193
CALCulate[:SELected]:MARKer:FUNCtion:RISetime <num> ............................................. 194
CALCulate[:SELected]:MARKer:FUNCtion:TARGet <value> ............................................. 194
CALCulate[:SELected]:MARKer:FUNCtion:TDIRection <value> ....................................... 195
CALCulate[:SELected]:MARKer<n>:FUNCtion:TRACking <bool> .................................... 196
CALCulate[:SELected]:MARKer:FUNCtion:ZONE <bool> .................................................. 197
CALCulate[:SELected]:MARKer<n>:SET <char> ............................................................... 197
CALCulate[:SELected]:MARKer<n>:SET:CENTer ............................................................. 198
CALCulate[:SELected]:MARKer<n>:SET:REFLevel .......................................................... 198
CALCulate[:SELected]:MARKer<n>[STATE] <char> .......................................................... 199
CALCulate[:SELected]:MARKer<n>:TRACe <tnum> ........................................................... 200
CALCulate[:SELected]:MARKer<n>:X <num> ................................................................. 200
CALCulate[:SELected]:MARKer<n>:Y? ............................................................................. 201
CALCulate[:SELected]:MARKer:Y<n>:UNCertainty? ......................................................... 202
CALCulate[:SELected]:MARKer:FUNCTION <char> .......................................................... 202
CALCulate[:SELected]:SMOothing:APERture <num> ....................................................... 202
CALCulate[:SELected]:SMOothing[:STATe] <bool> ......................................................... 203
CALCulate[:SELected]:TIME:AUX:CENTer <num> ............................................................ 204
CALCulate[:SELected]:TIME:AUX:PDIVision <num> ......................................................... 204
CALCulate[:SELected]:TIME:CENTer <num> ................................................................. 204
CALCulate[:SELected]:TIME:LENGth <num> ................................................................. 204
CALCulate[:SELected]:TIME:PDIVision <num> ............................................................... 204
CALCulate[:SELected]:TIME:STARt <num> ................................................................. 204
CALCulate[:SELected]:TRACe<n>:DATA? ........................................................................ 206
CALCulate[:SELected]:TRANSform:DISTance:BDPass <bool> ........................................ 207
CALCulate[:SELected]:TRANSform:DISTance:FREQuency:CENTer <num> ................... 207
CALCulate[:SELected]:TRANSform:DISTance:FREQuency:SPAN:MAXimum <num> ....... 207
CALCulate[:SELected]:TRANSform:DISTance:FREQuency:STARt:MINimum <num> ....... 208
::CHSCanner[:SENSe]:POWer[:RF]:GAIN[:STATe] ................................................................. 225
::CHSCanner:SWEep:AVERage:COUNt ............................................................................... 226
::CHSCanner:SWEep:AVERage[:STATe] ............................................................................. 226
::CHSCanner:SWEep:DISPlay:TYPE .................................................................................. 227
::CHSCanner:SWEep:MODE .............................................................................................. 227
::CHSCanner:USER:FOLDer ............................................................................................. 228
[:SENSe]:SALListen:DTYPE .............................................................................................. 228
[:SENSe]:SALListen:LTIMe ............................................................................................... 229
[:SENSe]:SALListen:PAUSE .............................................................................................. 229
SENSe:SALListen:RESume .............................................................................................. 230
:CONFigure? .................................................................................................................... 230
:CONFigure:SPECTrum .................................................................................................... 231
:CONFigure:WAVeform .................................................................................................. 231
::DISPlay:ANNotation:FREQuency <value> ................................................................. 232
::DISPlay:BRIGHTness <num> ....................................................................................... 233
::DISPlay:DATE:FMT <char> ........................................................................................ 233
::DISPlay:ENABle <bool> ............................................................................................. 234
::DISPlay:GRID <bool> .................................................................................................. 234
::DISPlay:KEYWord:DEFault .......................................................................................... 235
::DISPlay:KEYWord:[DATA] <string1,2,3,4,5,6,7,8> ...................................................... 235
::DISPlay:LTEFdd:BCHart[:SCALe]:AUTO ....................................................................... 235
::DISPlay:LTEFdd:BCHart[:SCALe]:PDIVision <num> ...................................................... 236
::DISPlay:LTEFdd:BCHart[:SCALe]:RLEVel ............................................................... 236
::DISPlay:LTEFdd:BCHart[:SCALe]:AUTO ........................................................................ 237
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 237
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 238
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 238
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 238
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 239
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 239
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 239
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 240
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 240
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 240
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 241
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 241
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 241
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 242
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 242
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 242
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 243
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 243
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 243
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 244
::DISPlay:LTEFdd:SCHart[:SCALe]:AUTO ........................................................................ 244
::DISPlay:LTEFdd:SCHart[:SCALe]:PDIVision ............................................................... 244
::DISPlay:LTEFdd:SCHart[:SCALe]:RLEVel ............................................................... 245
::DISPlay:MARKer:LARGE[:STATe] <char> ..................................................................... 245
::DISPlay:MARKer:LARGE[:x]:DEFIne:LINE<n>:FO RMat <char> ........................................ 245
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:LINE:MARKer&lt;n&gt;:STATe &lt;char&gt;</td>
<td>........................................... 246</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:LINE&lt;n&gt;:STATe &lt;bool&gt;</td>
<td>.................................................. 247</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:LINE&lt;n&gt;:TNUMber &lt;value&gt;</td>
<td>.................................................... 248</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:TRACe&lt;n&gt;:BWIDth:STATe &lt;value&gt;</td>
<td>....................................... 249</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:TRACe&lt;n&gt;:FORMat &lt;value&gt;</td>
<td>.................................................. 250</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:DEFine:TRACe&lt;n&gt;:MEASurement &lt;value&gt;</td>
<td>........................................ 250</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:FONT &lt;char&gt;</td>
<td>.......................................................... 251</td>
</tr>
<tr>
<td>DISPlay:MARKer:LARGE:&lt;x&gt;:TRACe:COUNt &lt;char&gt;</td>
<td>.................................................. 252</td>
</tr>
<tr>
<td>:DISPlay:SCReen:GEOMetry &lt;char&gt;</td>
<td>.......................................................... 252</td>
</tr>
<tr>
<td>DISPlay:TABLE:MARKer &lt;bool&gt;</td>
<td>.......................................................... 253</td>
</tr>
<tr>
<td>DISPlay:TABLE:RESults:DATA?</td>
<td>.......................................................... 253</td>
</tr>
<tr>
<td>DISPlay:TABLE:RESults &lt;bool&gt;</td>
<td>.......................................................... 254</td>
</tr>
<tr>
<td>DISPlay:TIME:FMT &lt;char&gt;</td>
<td>.......................................................... 254</td>
</tr>
<tr>
<td>DISPlay:TITLe:DATA &lt;string&gt;</td>
<td>.......................................................... 255</td>
</tr>
<tr>
<td>DISPlay:TITLe[:STATe] &lt;bool&gt;</td>
<td>.......................................................... 255</td>
</tr>
<tr>
<td>DISPlay:V5G:BCHart:Y[:SCALe]:AUTO</td>
<td>.......................................................... 255</td>
</tr>
<tr>
<td>DISPlay:V5G:BCHart:Y[:SCALe]:PDIVision</td>
<td>.......................................................... 256</td>
</tr>
<tr>
<td>DISPlay:V5G:BCHart:Y[:SCALe]:RLEVel</td>
<td>.......................................................... 256</td>
</tr>
<tr>
<td>DISPlay:V5G:SCHart:Y[:SCALe]:AUTO</td>
<td>.......................................................... 256</td>
</tr>
<tr>
<td>DISPlay:V5G:SCHart:Y[:SCALe]:PDIVision</td>
<td>.......................................................... 257</td>
</tr>
<tr>
<td>DISPlay:V5G:SCHart:Y[:SCALe]:RLEVel</td>
<td>.......................................................... 257</td>
</tr>
<tr>
<td>DISPlay:V5G:SPECtrum:Y[:SCALe]:AUTO</td>
<td>.......................................................... 258</td>
</tr>
<tr>
<td>DISPlay:V5G:SPECtrum:Y[:SCALe]:PDIVision</td>
<td>.......................................................... 258</td>
</tr>
<tr>
<td>DISPlay:V5G:SPECtrum:Y[:SCALe]:RLEVel</td>
<td>.......................................................... 259</td>
</tr>
<tr>
<td>DISPlay:V5G:SPECtrum:Y[:SCALe]:RPOSition</td>
<td>.......................................................... 259</td>
</tr>
<tr>
<td>DISPlay:V5G:WINDow&lt;n&gt;:CCARrier &lt;string&gt;</td>
<td>.......................................................... 259</td>
</tr>
<tr>
<td>DISPlay:V5G:WINDow&lt;n&gt;:DATA &lt;string&gt;</td>
<td>.......................................................... 260</td>
</tr>
<tr>
<td>DISPlay:V5G:WINDow&lt;n&gt;:SORT:DATA &lt;string&gt;</td>
<td>.......................................................... 261</td>
</tr>
<tr>
<td>DISPlay:V5G:WINDow&lt;n&gt;:SORT:ORDer &lt;string&gt;</td>
<td>.......................................................... 262</td>
</tr>
<tr>
<td>DISPlay:V5G:WINDow&lt;n&gt;:STATe</td>
<td>.......................................................... 262</td>
</tr>
<tr>
<td>DISPlay:V5G:TRACe:DATA &lt;string&gt;</td>
<td>.......................................................... 263</td>
</tr>
<tr>
<td>DISPlay:WAVeform:VIEW[:,:SElect] &lt;char&gt;</td>
<td>.......................................................... 263</td>
</tr>
<tr>
<td>:DISPlay:VIEW:DENSity:STATe</td>
<td>.......................................................... 264</td>
</tr>
<tr>
<td>:DISPlay:VIEW:DENSity:PERSistence</td>
<td>.......................................................... 264</td>
</tr>
<tr>
<td>:DISPlay:VIEW:DENSity:PERSistence:INFinite</td>
<td>.......................................................... 265</td>
</tr>
<tr>
<td>DISPlay:WINDow:ANALog:LOWer &lt;num&gt;</td>
<td>.......................................................... 265</td>
</tr>
<tr>
<td>DISPlay:WINDow:ANALog:UPPer &lt;num&gt;</td>
<td>.......................................................... 266</td>
</tr>
<tr>
<td>DISPlay:WINDow[NUMeric]:RESolution &lt;num&gt;</td>
<td>.......................................................... 266</td>
</tr>
<tr>
<td>DISPlay:WINDow[:,:SElect] &lt;int&gt;</td>
<td>.......................................................... 267</td>
</tr>
<tr>
<td>:DISPlay:WINDow:SPLit &lt;char&gt;</td>
<td>.......................................................... 267</td>
</tr>
</tbody>
</table>
DISPLAY:WINDow:TRACE<n>:MEMory[:STATE] <bool> .......................................................... 268
DISPLAY:WINDow:TRACE<n>:STATE <bool> ......................................................................... 268
DISPLAY:WINDow:TRACE<n>:STORE .................................................................................. 269
DISPLAY:WINDow:ZOOM <bool> ......................................................................................... 269
DISPLAY:WINDow:TRACE<n>:Y:DLIne <num> ............................................................... 270
DISPLAY:WINDow:TRACE<n>:Y:DLIne:STATE <bool> .................................................... 270
DISPLAY:WINDow:TRACE<n>:MEMory[:STATE] <bool> ........................................... 271
DISPLAY:WINDow:TRACE<n>:STATE <bool> ......................................................................... 271
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:AUTO ................................................................. 272
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:BOTTOM <num> ............................................... 273
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:PDIVision <num> .............................................. 273
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:RLEVEL <num> .................................................. 274
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:RPOSITION <num> ........................................... 275
DISPLAY:WINDow:TRACE<n>:Y[:SCALE]:TOP <num> ....................................................... 275
DISPLAY:WINDow<n>:DATA <char> ................................................................................ 276
DISPLAY:WINDow<n>:STATE <int> .................................................................................. 277
FORMAT:BORDER <char> ............................................................................................... 278
FORMAT[:DATA] <char> .......................................................................................... 278
INITiate:SIQCapture .................................................................................................. 280
INITiate:CONTinuous <bool> ......................................................................................... 280
INITiate:[IMMediate] .................................................................................................. 281
INITiate:RESTart ......................................................................................................... 281
:INPut:LAN:IDENtify:SNUMber <number> ................................................................. 282
:INPut:LAN:IDENtify:TYPE SNUM | IPA .................................................................. 283
INSTRument:CATalog? .................................................................................................. 283
INSTRument:GTR ........................................................................................................ 284
INSTRument:[SELect] <string> .................................................................................... 284
:LTEFdd:DATA? ........................................................................................................ 285
MMEMory:CDIRectory <string> ..................................................................................... 286
MMEMory:COPY <file1>,<file2> ................................................................................. 287
MMEMory:DATA <filename>,<data> ............................................................................. 288
MMEMory:DELeTe <string> ........................................................................................ 288
MMEMory:IMPort:DUT <string>,<char> .................................................................... 289
MMEMory:IMPort:PAMPlifier <string>,<char> ............................................................ 289
MMEMory:LOAD:PAMPlifier <string>,<char> ............................................................. 290
MMEMory:LOAD:ANTeena <string>,<char> .............................................................. 290
MMEMory:LOAD:CABLE <string>,<char> ................................................................. 291
MMEMory:LOAD:DUT <string>,<char> ..................................................................... 292
MMEMory:LOAD:ENR <string>,<char> ..................................................................... 293
Table of Contents

:MMEMory:LOAD:LIST .............................................................................................................................. 293
:MMEMory:LOAD:OTA:LOG <string> .......................................................................................................... 294
:MMEMory:LOAD:LOG ................................................................................................................................. 294
MMEMory:LOAD:STATE <string> .............................................................................................................. 295
MMEMory:MDIRectory <string> ................................................................................................................. 295
MMEMory:MOVE <file1,<file2> ................................................................................................................. 296
MMEMory:RDIRectory <string> [.<recursive>] .......................................................................................... 296
:MMEMory:STATE:STORE:LIST ................................................................................................................... 297
MMEMory:STORE:ANTenna <string>,<char> ............................................................................................. 298
MMEMory:STORE:CABLE <string>,<char> ............................................................................................... 298
MMEMory:STORE:DUT <string>,<char> ..................................................................................................... 299
MMEMory:STORE:ENR <string>,<char> ..................................................................................................... 300
MMEMory:STORE:FDATA <filename> ........................................................................................................ 300
:MMEMory:STORE:IMAGE <string> .......................................................................................................... 301
:MMEMory:STORE:IMAGE:NOKeys <string> ............................................................................................ 301
MMEMory:STORE:IQCapture:DEVICE ....................................................................................................... 302
MMEMory:STORE:IQCapture:FCOUNT ....................................................................................................... 302
MMEMory:STORE:IQCapture:FOUnt:MULTIPLE <bool> ............................................................................... 303
MMEMory:STORE:IQCapture:FNAMe <string> .......................................................................................... 303
MMEMory:STORE:IQCapture:FTYPE <char> ............................................................................................. 304
MMEMory:STORE:IQCapture:START ......................................................................................................... 304
MMEMory:STORE:IQCapture:STOP ............................................................................................................ 305
:MMEMory:STORE:LOG:CSV ...................................................................................................................... 305
:MMEMory:STORE:LOG:KML ..................................................................................................................... 306
MMEMory:STORE:MAT <string> ................................................................................................................ 306
MMEMory:STORE:PAMPLifier <string>,<char> .......................................................................................... 306
MMEMory:STORE:SDF <string> .................................................................................................................. 307
MMEMory:STORE:SNP[:DATA] <filename> ............................................................................................... 308
MMEMory:STORE:STATE <string> ............................................................................................................ 308
MMEMory:STORE:TXN <string>,<char> .................................................................................................... 308
:RECPlayback:ACTION:PAUSE .................................................................................................................. 309
:RECPlayback:ACTION:PLAY ...................................................................................................................... 309
:RECPlayback:ACTION:POSITION <num> .............................................................................................. 310
:RECPlayback:ACTION:POSITION:AUTO <string/num> ........................................................................ 310
:RECPlayback:ACTION:RECORD ................................................................................................................ 310
:RECPlayback:ACTION:SAVE ................................................................................................................... 311
:RECPlayback:ACTION:SPOSITION <num> .............................................................................................. 311
:RECPlayback:ACTION:STOP ..................................................................................................................... 312
:RECPlayback:ACTION:TPOSITION <num> .............................................................................................. 312
:RECPlayback:CONFIG:FILE:DEVICE <string> ....................................................................................... 312
:RECPlayback:CONFIG:FILE:OWRITE <string/boolean> ........................................................................ 313
:RECPlayback:CONFIG:FILE:TYPE <string> .......................................................................................... 314
[:SENSe]:CORRection:COLLect[:ACQuire]:THRU <p1>,<p2> ............................................................. 356
[:SENSe]:CORRection:COLLect:CKIT:LABel <port>,<ckit> ................................................................. 356
[:SENSe]:CORRection:COLLect:CKIT:LABel:CATalog? ...................................................................... 358
[:SENSe]:CORRection:COLLect:CONNector <port>,<type -gen-,imp> ................................................ 358
[:SENSe]:CORRection:COLLect:ECAL:AORient <bool> ...................................................................... 359
[:SENSe]:CORRection:COLLect:ECAL:SIMPle <bool> ........................................................................ 360
[:SENSe]:CORRection:COLLect:GUIDed:SCOunt? ............................................................................. 360
[:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire <num> ....................................................... 361
[:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMpt? <num> ..................................................... 361
[:SENSe]:CORRection:COLLect:METHod:ERES <p1>,<p2> ............................................................... 362
[:SENSe]:CORRection:COLLect:METHod:ORESponse <p1>.............................................................. 362
[:SENSe]:CORRection:COLLect:METHod:QCALibrate:CALibrate <p1>,<p2> ....................................... 363
[:SENSe]:CORRection:COLLect:METHod:QCALibrate:ERESponse <p1>,<p2> ................................. 363
[:SENSe]:CORRection:COLLect:METHod:QSOLt <p1,p2> .................................................................. 364
[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:OPEN <p1> ................................................... 365
[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:SHORt <p1> ................................................. 366
[:SENSe]:CORRection:COLLect:METHod:METHod:SOLT1 <p1> ........................................................... 367
[:SENSe]:CORRection:COLLect:METHod:METHod:SOLT2 <p1,p2> .................................................. 367
[:SENSe]:CORRection:COLLect:METHod:METHod:SOLR <p1>,<p2>. .................................................. 368
[:SENSe]:CORRection:COLLect:METHod:METHod:SRESponse <p1> ................................................ 369
[:SENSe]:CORRection:COLLect:METHod:METHod:TRL <p1,p2> .......................................................... 369
[:SENSe]:CORRection:COLLect:METHod:METHod:TYPE? ............................................................... 370
[:SENSe]:CORRection:COLLect:OISolation <bool> ............................................................................. 370
[:SENSe]:CORRection:COLLect:SOLR <p1>,<p2> ............................................................... 368
[:SENSe]:CORRection:COLLect:SOLR <p1>,<p2> ............................................................... 368
[:SENSe]:CORRection:COLLect:SRESponse <p1> .............................................................. 369
[:SENSe]:CORRection:COLLect:SAVE <num> .................................................................................... 371
[:SENSe]:CORRection:DUT:MODe <char> .......................................................................................... 371
[:SENSe]:CORRection:DUT:SPOT:DISTRIBUTion .............................................................................. 372
[:SENSe]:CORRection:DUT:SPOT:OUTGamma .................................................................................. 373
[:SENSe]:CORRection:DUT:SPOT:SPECify ..................................................................................... 373
[:SENSe]:CORRection:ENR:EXTRapolated? ....................................................................................... 375
[:SENSe]:CORRection:ENR:MODe <char> .......................................................................................... 375
[:SENSe]:CORRection:ENR:SPOT:COVerage <char> .......................................................................... 376
[:SENSe]:CORRection:ENR:SPOT:DISTRIBUTion <char> ............................................................... 376
[:SENSe]:CORRection:ENR:SPOT:ENR <num> .................................................................................. 377
[:SENSe]:CORRection:ENR:SPOT:OFF:Gamma <num> .............................................................. 378
[:SENSe]:CORRection:ENR:SPOT:ONGamma <num> ........................................................................ 378
[:SENSe]:CORRection:ENR:SPOT:SPEC <char> ................................................................................. 379
[:SENSe]:CORRection:ENR:SPOT:UNCertainty <num> .................................................................... 380
[:SENSe]:CORRection:EXTension:PORT1 <num> ............................................................................. 380
[:SENSe]:CORRection:EXTension:PORT2 <num> ............................................................................. 381
[:SENSe]:CORRection:EXTension[:STATe] <bool> ........................................................................... 381
Table of Contents

[SENSe]:CORRection:GAIN2:INPut:MAGNitude <num> ................................................................. 382
[SENSe]:CORRection:GAIN2:STATe <bool> ................................................................................ 382
[SENSe]:CORRection:IMPedance:INPut:MAGNitude <num> ......................................................... 383
[SENSe]:CORRection:LOSS:AFTer:ENABled <bool> ..................................................................... 383
[SENSe]:CORRection:LOSS:AFTer:VALue <double> ................................................................. 384
[SENSe]:CORRection:LOSS:BEFore:ENABled <bool> ................................................................. 384
[SENSe]:CORRection:LOSS:BEFore:VALue <double> .............................................................. 385
[SENSe]:CORRection:LOSS:COAX <num> .................................................................................. 385
[SENSe]:CORRection:LOSS:WAVeguide ...................................................................................... 386
[SENSe]:CORRection:MEDium <char> ...................................................................................... 386
[SENSe]:CORRection:PAMPLifier:MODe <char> ....................................................................... 386
[SENSe]:CORRection:PAMPLifier:SPOT:DISTRIBUTion <char> ................................................... 387
[SENSe]:CORRection:PAMPLifier:SPOT:INGamma <num> ....................................................... 388
[SENSe]:CORRection:PAMPLifier:SPOT:OUTGamma <num> .................................................... 388
[SENSe]:CORRection:PAMPLifier:SPOT:SPEC <char> ............................................................. 389
[SENSe]:CORRection:RCALibration:CANCel .............................................................................. 390
[SENSe]:CORRection:RCALibration:RUN ................................................................................... 390
[SENSe]:CORRection:RCALibration:STATe]? ............................................................................ 391
[SENSe]:CORRection:RLEVELocity:COAX <num> .................................................................... 391
[SENSe]:CORRection:[STATe] <bool> ...................................................................................... 391
[SENSe]:CORRection:TCOLd <num> .......................................................................................... 392
[SENSe]:CORRection:UCALibration:CANCel .............................................................................. 393
[SENSe]:CORRection:UCALibration:INTERpolated? .................................................................... 394
[SENSe]:CORRection:UCALibration:RUN .................................................................................. 394
[SENSe]:CORRection:UCALibration:STATe <bool> .................................................................. 395
[SENSe]:CORRection:UNCertainty:BARS .................................................................................. 395
[SENSe]:CORRection:UNCertainty:CALibration <bool> .......................................................... 396
[SENSe]:CORRection:UNCertainty:COVerage <char> .............................................................. 396
[SENSe]:CORRection:UNCertainty:ENR<bool> ......................................................................... 397
[SENSe]:CORRection:UNCertainty:JITTER <bool> ................................................................. 398
[SENSe]:CORRection:UNCertainty:MIISMatch <bool> ............................................................. 398
[SENSe]:CORRection:UNCertainty:QUEStionable? ................................................................. 399
[SENSe]:CORRection:WAVeguide:STANDard <char> ............................................................... 399
[SENSe]:CORRection:USER:STATe <bool> .................................................................................. 400
[SENSe]:CORRection:WGCutoff <num> ..................................................................................... 400
[SENSe]:CORRection:ZERO:REFerence <char> ......................................................................... 401
[SENSe]:CORRection:ZERO:STATe <char> ................................................................................. 401
[SENSe]:DIF:BANDwidth <num> .............................................................................................. 402
[SENSe]:DENSity:BPLevel ........................................................................................................ 402
[SENSe]:DENSity:RPLevel .......................................................................................................... 403
[SENSe]:DETe ctor:FUNCtion <char> ...................................................................................... 403
[SENSe]:DETe ctor:TRAcE{1:4}:FUNCtion ............................................................................. 404
[:SENSe]:SPECTrum:BANDwidth[:RESolution] <num> ................................................................. 446
[:SENSe]:SPECTrum:BANDwidth[:RESolution]:AUTO <bool> .................................................... 447
[:SENSe]:SPECTrum:FFT:ANALysis:LENGTH <num> ................................................................. 447
[:SENSe]:SPECTrum:FFT:LENGTH:AUTO <bool> ................................................................. 448
[:SENSe]:SPECTrum:FFT:WINDow <char> ................................................................................. 448
[:SENSe]:SPECTrum:FREQuency:SPAN <freq> ................................................................. 449
[:SENSe]:SWEep:ACQuisition <num> .................................................................................. 449
[:SENSe]:SWEep:ACQuisition:AUTO <bool> ........................................................................ 450
[:SENSe]:SWEep:MTIME? ........................................................................................................ 450
[:SENSe]:SWEep:POINTs <num> ............................................................................................ 451
[:SENSe]:SWEep:RX <value> .................................................................................................. 451
[:SENSe]:SWEep:TIME <num> .................................................................................................. 452
[:SENSe]:SWEep:TYPE <char> .................................................................................................. 452
[:SENSe]:SWEep:TYPE <value> .................................................................................................. 453
[:SENSe]:TAListen:AVOLume <num> - Obsolete ...................................................................... 453
[:SENSe]:TAListen:DSTate <bool> .......................................................................................... 454
[:SENSe]:TAListen:DTYPe <char> .......................................................................................... 454
[:SENSe]:TAListen:LTIMe <num> .......................................................................................... 455
[:SENSe]:TAListen:TFReq <num> .......................................................................................... 455
[:SENSe]:TOL <value> ............................................................................................................. 456
[:SENSe]:TRACe[:DATA]? ........................................................................................................ 456
[:SENSe]:TRACe:LIMIT:LOWer <num> ................................................................................ 456
[:SENSe]:TRACe:LIMIT:LOWer:STATe <bool> ....................................................................... 457
[:SENSe]:TRACe:LIMIT:UPPer <num> .................................................................................. 457
[:SENSe]:TRACe:LIMIT:UPPer:STATe <bool> ....................................................................... 458
[:SENSe]:TRACe:MEASurement:REFerence <num> ............................................................ 458
[:SENSe]:TRACe:MEASurement <value> ............................................................................. 459
[:SENSe]:V5G:CCARrier<n>:ENABle .................................................................................... 459
[:SENSe]:V5G:CCARrier<n>:FREQuency:CENTer ..................................................................... 460
[:SENSe]:V5G:SCORr:THReshold <num> ............................................................................... 460
[:SENSe]:WAveform:START <num> .................................................................................... 461
[:SENSe]:WAveform:STOP <num> ...................................................................................... 462
SOURce:ENABle <bool> .......................................................................................................... 462
SOURce:ENABle <bool> .......................................................................................................... 463
SOURce:FREQuency:CENTer <value> .................................................................................. 463
SOURce:FREQuency:[CW] <num> ..................................................................................... 464
SOURce:FREQuency:SPAN <value> .................................................................................... 464
SOURce:FREQuency:START <value> .................................................................................. 465
SOURce:FREQuency:STOP <value> ..................................................................................... 466
SOURce:MODE <char> ............................................................................................................ 466
SOURce:NORMalize <bool> .................................................................................................. 467
SOURce:POWer <num> ............................................................................................................ 467
SYSTem:GPS:DISPLAY:DISTANCE:UNIT <char> ................................................................. 483
SYSTem:GPS:DISPLAY:STATE <bool> .............................................................................. 484
SYSTem:GPS:LSState? ........................................................................................................ 484
SYSTem:GPS:[STATE] <char> .................................................................................................. 485
SYSTem:GPS:SYNChronize <bool> .................................................................................. 485
SYSTem:PREFerences:DFLT ............................................................................................ 486
SYSTem:PREFerences:SAVE ........................................................................................... 486
:SYSTem:PRESet ................................................................................................................ 486
:SYSTem:PRESet:MODE ................................................................. 486
SYSTem:PWR:AUTO <value> ......................................................................................... 487
SYSTem:PWR:SHUTdown <value> .................................................................................... 487
SYSTem:PWR:SHUTdown:DLY <value> ........................................................................... 487
SYSTem:PWR:SHUTdown:DURATION <value> ................................................................. 487
SYSTem:PWR:SUSP <bool> .............................................................................................. 488
SYSTem:PWR:SUSP:DLY <value> .................................................................................... 488
SYSTem:PWR:SUSP:DURATION <value> ........................................................................... 489
SYSTem:UPReset:FPANel[:STATE] <bool> ................................................................. 489
SYSTem:UPReset:MODE ............................................................................................ 489
SYSTem:UPReset:SAVE ........................................................................................... 489
SYSTem:UPReset ................................................................................................................ 489
SYSTem:TIME <hr, min, sec> ............................................................................................ 490
SYSTem:TZONe <string> .................................................................................................. 490
SYSTem:TZONe:CATalog? .............................................................................................. 490
SYSTem:VERSion? ........................................................................................................ 491
SYSTem:VVS:CURRENT? .............................................................................................. 491
SYSTem:VVS:ENABLE <bool> ....................................................................................... 492
SYSTem:VVS:HIMD <bool> ......................................................................................... 492
SYSTem:VVS:MAXCurrent? ............................................................................................. 492
SYSTem:VVS:MVOLTage? .................................................................................................. 493
SYSTem:VVS:RVOltage <num> .......................................................................................... 493
SYSTem:VVS:[STATE]? .................................................................................................... 493
SYSTem:VVS:VOLTage <num> ........................................................................................... 493
TRAcE<n>:DATA? <char> ............................................................................................... 495
TRAcE:ERTA:RINPut <char> ............................................................................................ 495
TRAcE:IMAGinary:DATA? .............................................................................................. 496
:TRAcE:IMAGinary:Y:AUTO ............................................................................................. 497
:TRAcE:IMAGinary:Y:DLINE <volt> ............................................................................... 497
:TRAcE:IMAGinary:Y:PDIVision <volt> ........................................................................ 497
:TRAcE:IMAGinary:Y:RLEVel <volt> .............................................................................. 497
:TRAcE:IMAGinary:Y:RPOSITION <int> ........................................................................ 497
TRAcE:IQCapture:DATA? ............................................................................................. 500
TRAcE<n>:JITter:WARNing:DATA? ................................................................................. 500
Table of Contents

TRIGger:SOURce <char> ..................................................................................................................... 543
UNIT:POWer <string> ...................................................................................................................... 544
:V5G:DATA:GPS? ................................................................................................................................. 544
:V5G:DATA? .......................................................................................................................................... 544
FieldFox Programming Guide
All Models

• Commands Common to All Modes

• List of Commands by Mode
  o CAT Mode
  o NA Mode
  o SA Mode
  o RTSA Mode
  o IQA Mode
  o VVM Mode
  o USB Power Meter Mode
    o FOPS
  o Pulse Measurements
  o CPM Mode
  o ERTA Mode
  o AM/FM Metrics
  o Channel Scanner Mode
  o OTA:
    o 5GTF NEW!
    o LTE FDD NEW!
  o Noise Figure

See Also

• Example Programs
• SCPI Concepts and Tips
• New Commands with this release.
• Status Registers
• Calibration Commands
• Instrument Console
• FieldFox User’s Guides

• Command Reference - See Table of Contents

Supports Firmware Revision: A.08.18 and A11.00
Date: 2018-12-19
FieldFox Programming Tips

Mode-specific Programming
Although the FieldFox is a single instrument, each FieldFox mode (NA, SA, and so forth) has its own unique set of SCPI commands. Each mode is targeted separately by first issuing the command to select the mode (INSTRument[:SELection]).

Here is a list of commands that are common to ALL modes: Commands Common to All Modes.

Perform Single Triggering
When programming the FieldFox, it is ALWAYS recommended to perform single sweep triggering: INITiate:CONTinuous 0 and INITiate[:IMMediate] followed with *OPC?.

This is because after making measurement settings such as setting frequency, there is NO guarantee that a continuous sweep will complete and data will be collected at the new setting. By following a series of settings with INIT:IMM;*OPC?, then all settings will be updated correctly.

Communicating with the FieldFox using sockets over LAN
Responses to SCPI commands will always be in ASCII string format unless otherwise noted. Long responses may be separated into packets of data (up to 1460 bytes long). Each response is terminated with a LF character. When receiving long responses, search for the LF character to determine that the response is complete.

Do NOT do Binary Block transfers (FORM:DATA REAL,32 or REAL,64) when using over Telnet to port 5024 on FieldFox.
About Calibration Settings

A calibration session that is performed using the front-panel is completely separate from a calibration session that is performed programmatically.

Therefore, calibration settings that are made remotely (such as setting the cal kit and connectors) can NOT be observed from the front-panel user interface. Alternatively, calibration settings are that are made from the FieldFox front panel (user interface) can NOT be queried programmatically.

Correction Methods Explained

The Correction (Calibration) Methods offered for the FieldFox allow you to balance higher accuracy or a faster sweep time. Several methods are available through SCPI that are NOT available using the front-panel user interface. For example, all of the Cal Methods normally available only in NA mode are also available programmatically in CAT modes.

- Definitions
- Cal Methods
  - 1-Port (OSL)
  - FULL 2-Port
  - SOLR
  - QSOLT
  - Enhanced Response Cal
  - Response Cals
- FieldFox Model Summary

See Also
- Calibration Examples
- CAT Mode Commands
- NA Mode Commands

Definitions

Non-insertable DUT – A device whose connectors could NOT mate together. They either do not have the same type of connector or they have the same gender. This also means that the test port cables could NOT mate together as in the above diagram.

Insertable DUT – A device whose connectors could mate together. They have the same type of connector and opposite or no gender. This also means that the test port cables could mate together, as in the above diagram.

Flush THRU - When the test port cables mate together when measuring an Insertable DUT. The THRU standard has no loss and no electrical length.
Sweep Directions – Relevant to N9923A ONLY.

- Sweep in FORWARD direction means port 1 is the source port and port 2 is the receiver port. Used to measure S11 and S21. (N9912A sweeps in forward direction ONLY)
- Sweep in REVERSE direction means port 2 is the source port and port 1 is the receiver port. Used to measure S22 and S12.

Full 2-Port and QSOLT Cals result in correction that requires background measurement sweeps in both directions, regardless of the displayed measurements. The displayed traces are updated at a slower rate than Enhanced Response and 1-port cals, which require sweeps in one direction only.

Cal Methods

1-Port (OSL)
- Insertability - Not Relevant
- S-parameters Corrected: S11 or S22 (N9923)
- Standards: OPEN, SHORT, LOAD
- Sweeps in ONE direction.

FULL 2-Port (N9923A with Opt 122)
- Mechanical or QuickCal (Opt 111 or 112)
- Corrects all S-parameters.
- Non-Insertable or Insertable DUT
- Standards: OPEN, SHORT, LOAD on BOTH ports. Known (characterized) THRU between ports.
- Sweeps in BOTH directions.

Note: Because FULL 2-Port method requires a known THRU connection between the test ports, it is better to use SOLR, which yields the same level of accuracy with ANY (unknown) THRU connection. SOLR is performed when calibrating both ports from the front-panel user interface.

SOLR (Short-Open-Load-Reciprocal Thru) (All models with Full 2-port option)
Also known as Unknown Thru calibration.
- Mechanical Cal ONLY
- Most comprehensive calibration. Corrects all S-parameters.
- Non-Insertable or Insertable DUT
- Standards: OPEN, SHORT, LOAD on BOTH ports. Any THRU between ports.
- Sweeps in BOTH directions.

QSOLT (All models with Full 2-port option)
- Mechanical Cal ONLY
- Quicker to perform than Full 2-port. Corrects all S-parameters.
- Insertable DUT only
- Standards: OPEN, SHORT, LOAD on ONE port. Flush THRU between ports.
- Sweeps in BOTH directions.

Enhanced Response Cal
- Forward (all models) OR Reverse (All models with Full 2-port option)
- Mechanical or QuickCal (Opt 111 or 112)
- Faster measurements than Full 2-Port because sweeps in one direction ONLY.
- Insertable DUT only
- S-parameters Corrected: S21 and S11 (Forward) OR S12 and S22 (Reverse)
• Standards: OPEN, SHORT, LOAD on ONE port. Flush THRU between ports.

**Response Cals**
- Least accurate Cal type - correct Magnitude ONLY.
- Mechanical Cal ONLY
- Perform same as Data/Memory or THRU Normalization
- Standards: OPEN or SHORT (1-port) or THRU (2-port)

**Summary**

**N9912A - CAT and NA**

<table>
<thead>
<tr>
<th>Cal Methods</th>
<th>Command click to see command</th>
<th>Ports &lt;p&gt;</th>
<th>Req'd Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuickCals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-port OSL</td>
<td>QCAL:CAL &lt;p&gt;</td>
<td>1</td>
<td>111</td>
</tr>
<tr>
<td>Enhanced Response</td>
<td>QCAL:ERES &lt;p&gt;</td>
<td>1,2</td>
<td>111, 110</td>
</tr>
</tbody>
</table>

**Mechanical Cals** - specify connector and cal kit

| 1-port OSL       | SOLT1 <p>                     | 1         | None          |
| Enhanced Response| ERES <p>                      | 1,2       | 110           |

**Response Cals**

| Open Response    | OPEN <p>                      | 1         | None          |
| Short Response   | SHORT <p>                     | 1         | None          |
| Thru Response    | THRU <p>                      | 1,2       | 110           |

**N9912A Options**
- CAT Mode - Standard
- Option 110 - Adds 2-port (Fwd) measurements
- Option 111 - Adds QuickCal
- Option 303 - Adds NA Mode

---

**All models with Full 2-port option - CAT and NA**

<table>
<thead>
<tr>
<th>Cal Methods</th>
<th>Command click to see command</th>
<th>Ports &lt;p&gt;</th>
<th>Req'd Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuickCals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-port OSL or Full 2 port</td>
<td>QCAL:CAL &lt;p&gt;</td>
<td>1 (1-port)</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 (1-port)</td>
<td>122,112</td>
</tr>
<tr>
<td></td>
<td>QCAL:ERES &lt;p&gt;</td>
<td>1.2 (Fwd)</td>
<td>112</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>Enhanced Response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-port OSL</td>
<td>SOLT1 &lt;p&gt;</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>Full 2 port</td>
<td>SOLT2 &lt;p&gt;</td>
<td>1,2</td>
<td>122, 211 1</td>
</tr>
<tr>
<td>Unknown Thru</td>
<td>SOLR &lt;p&gt;</td>
<td>1,2</td>
<td>122, 211 1</td>
</tr>
<tr>
<td>QSOLT</td>
<td>QSOLT &lt;p&gt;</td>
<td>1,2 or 2,1</td>
<td>122, 211 1</td>
</tr>
<tr>
<td>Enhanced Response</td>
<td>ERES &lt;p&gt;</td>
<td>1,2</td>
<td>None</td>
</tr>
<tr>
<td>2,1</td>
<td></td>
<td>122</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Cals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Response</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Short Response</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Thru Response</td>
</tr>
</tbody>
</table>

1. N9923A requires Option 122. For FieldFox N9914/5/6/7/8A and N995xA models with firmware versions ≥A.07.00, Option 211 is required. Option 211 requires Option 210.

**N9923A Options**
- NA Mode - Standard
- Option 112 - Adds QuickCal
- Option 122 - Adds Full 2-port (Fwd and Rev) Measurements
- Option 305 - Adds CAT Mode

**Options for N9914/5/6/7/8A and N995xA models**
- Option 211 - Adds Full 2-port (Fwd and Rev) Measurements - Requires Option 210
- Option 210 - Adds Vector Network Analyzer (VNA) with transmission/reflection (T/R) capability

**Options for All other N992x models**
- Option 112 - Adds QuickCal
- Option 211 - Adds Full 2-port (Fwd and Rev) Measurements
- Option 305 - Adds CAT Mode

**How the FieldFox Error Queue Works**
Errors work as follows:

1. Errors are logged to the central logger, visible with the FieldFox GUI. Press **System 7** then **Service Diagnostics**, then **Error Log**.

2. Errors originating from a specific client (connected program) are logged to the queue for that client. Each client has its own queue. So if you have two socket connections (not recommended), and the first one sends a bad command, the second one does NOT get a 'Header not found' error logged in its queue. Only the first one will see that error on **SYST:ERR?**

3. 'Global' errors (those that occur independent of a connected client), are logged to the central logger (per #1) and to all currently connected SCPI client error queues.

4. SCPI socket (port 5025) and Telnet (port 5024) clients are independent and dynamic. If there is no current connection (or connections), then no error queue for that connection (or connections) exist(s), and no 'Global' errors are logged there because there is no place to log them.

However, the VXI parser always exists in FieldFox firmware even if no clients are connected to it. This is the connection typically used by Keysight I/O Libraries (unless 'socket' connection is checked). So it will always queue 'Global' errors that occur, and a client that connects via the VXI interface after the errors occur may still query out those errors.

You can bind a C# program to the C# VISA wrapper provided with Keysight I/O Libraries in order to use the VXI interface to FieldFox, if you plan to connect after the errors occur to query them out of the FieldFox.

In cases where a persistent error is occurring (like ADC Overload), currently, the error queue for the VXI parser (which receives these errors, as mentioned above) is unbounded, and will accumulate errors until instrument memory is exhausted. This occurs unless a VXI client connects and clears the VXI parser's error queue (using SYST:ERR?).

**Instrument Console (IC)**

IC is a free utility that can be used to run simple example programs to control the FieldFox.

This utility is NOT supported. Use 'As-is'.


Unzip, then double-click on IC.exe to install.

**Note:** IC requires .NET (version 2.0 or later). If IC does NOT run, then download .NET at: [http://www.microsoft.com/net/download.aspx](http://www.microsoft.com/net/download.aspx)

**Hint**

IC can be used with IO libraries or by directly connecting to an IP address.

If an error is continuously returned, try increasing the timeout. (-t <time in seconds>).

Then reconnect by issuing -a <address>

**IC Command Summary:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[?</td>
<td>help]</td>
</tr>
<tr>
<td>-a &lt;address&gt;</td>
<td>sets a new address, e.g. 192.168.1.1</td>
</tr>
<tr>
<td>-d</td>
<td>detach from currently connected instrument.</td>
</tr>
<tr>
<td>-clear</td>
<td>Clear IO stream. Experimental.</td>
</tr>
<tr>
<td>-r</td>
<td>re-attach to currently connected instrument.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-t&lt;?</td>
<td>returns current timeout</td>
</tr>
<tr>
<td>-t&lt; time in secs&gt;</td>
<td>sets a new timeout, e.g. &quot;-t .010&quot; for 10ms</td>
</tr>
<tr>
<td>-w &lt;time in msecs&gt;</td>
<td>wait (pause) execution for the specified amount of time</td>
</tr>
<tr>
<td>-err[-?]</td>
<td>set/remove/query automatic SYST:ERR? after sending a command/query string</td>
</tr>
<tr>
<td>-visa?</td>
<td>List VISA instruments and aliases.</td>
</tr>
<tr>
<td>-alias x=[y]</td>
<td>Adds an alias 'x' for instrument name 'y'</td>
</tr>
<tr>
<td></td>
<td>If y is omitted, alias x is cleared.</td>
</tr>
<tr>
<td>-alias?</td>
<td>Returns existing aliases.</td>
</tr>
<tr>
<td>-sp &lt;script path&gt;</td>
<td>Sets ';' delimited path to locate scripts.</td>
</tr>
<tr>
<td>-sp?</td>
<td>Returns the script path.</td>
</tr>
<tr>
<td>-p [Message]</td>
<td>Pauses execution, prints optional Message, user must press return to</td>
</tr>
<tr>
<td></td>
<td>continue.</td>
</tr>
<tr>
<td>-v</td>
<td>-verify</td>
</tr>
<tr>
<td></td>
<td>where n=0 is most recent result (0&lt;=n&lt;=9)</td>
</tr>
<tr>
<td></td>
<td>e.g. &quot;-v $0==1&quot; will pass after a *OPC?</td>
</tr>
<tr>
<td></td>
<td>IC will exit with -1 if verify test fails.</td>
</tr>
<tr>
<td>!&lt;string&gt;</td>
<td>performs ShellExecute on &lt;string&gt;</td>
</tr>
<tr>
<td>$&lt;filename&gt;</td>
<td>opens filename and executes all lines of text</td>
</tr>
<tr>
<td></td>
<td>Lines beginning with # in the 1st column are treated as comments</td>
</tr>
<tr>
<td>&lt;string&gt;</td>
<td>sends command to the instrument</td>
</tr>
<tr>
<td>&lt;string&gt;??[args]</td>
<td>sends a command, then reads back data</td>
</tr>
<tr>
<td>[exit</td>
<td>quit]</td>
</tr>
</tbody>
</table>

Alpha support has been added for I/O redirection so that data can be read to/from files as part of sending command strings.

E.g.: SENS:FREQ:STAR < freq.txt where freq.txt contains a string such as "3e9"

or

SENS:FREQ:STAR? > freq.txt to dump the start frequency to freq.txt

This program also allows binaries to be properly redirected to a file, e.g. to save and get a screenshot PNG file, you can now do this (and it works without corrupting the PNG file):

MMEM:STOR:IMAG "my.png" this saves a png file on the FieldFox.

MMEM:DATA? "my.png" > my.png this transfers the file to the PC.
Most of the commands used for controlling instruments on the GPIB are SCPI commands. The following sections will help you learn to use SCPI commands in your programs.

- **Branches on the Command Tree**
- **Command and Query**
- **Multiple Commands**
- **Command Abbreviation**
- **Bracketed (Optional) Keywords**
- **Vertical Bars (Pipes)**
- **MIN and MAX Parameters**

### Branches on the Command Tree

All major functions on the analyzer are assigned keywords which are called ROOT commands. Under these root commands are branches that contain one or more keywords. The branching continues until each analyzer function is assigned to a branch. A root command and the branches below it is sometimes known as a subsystem.

For example, under **CALCulate** are several branch commands.

Sometimes the same keyword, such as **STATE**, is used in several branches of the command tree. To keep track of the current branch, the analyzer’s command parser uses the following rules:

- **Power On and Reset** - After power is cycled or after `*RST`, the current path is set to the root level commands.
- **Message Terminators** - A message terminator, such as a `<NL>` character, sets the current path to the root command level. Many programming language output statements send message terminators automatically.
- **Colon (:)** - When a colon is between two command keywords, it moves the current path down one level in the command tree. For example, the second colon in `:SOURCE:POWER` specifies that `POWER` is one level below `SOURCE`. When the colon is the first character of a command, it specifies that the following keyword is a root level command. For example, the first colon in `:SOURCE:POWER` specifies that source is a root level command.

**Note:** You can omit the leading colon if the command is the first of a new program line. For example, the following two commands are equivalent:

```
SOUR:POW:ATT:AUTO
:SOUR:POW:ATT:AUTO
```

- **Whitespace** - Whitespace characters, such as `<tab>` and `<space>`, are generally ignored. There are two important exceptions:
  - Whitespace inside a keyword, such as `:CALC ULATE`, is not allowed.
  - Most commands end with a parameter. You must use whitespace to separate these ending parameters from commands. **Always refer to the command documentation.** In the following example, there is whitespace between `STATE` and `ON`.

```
CALCULATE:LIMit:STATE ON
```

- **Comma (,)** - If a command requires more than one parameter, you must separate adjacent parameters using a comma. For example, the `SYSTEM:TIME` command requires three values to set the analyzer clock: one for hours, one for minutes, and one for seconds. A message to set the clock to 8:45 AM would be `SYSTEM:TIME 8,45,0`. Commas do not affect the current path.

- **Semicolon(;)** - A semicolon separates two commands in the same message without changing the current path. See **Multiple Commands** later in this topic.
IEEE 488.2 Common Commands - Common commands, such as \*RST, are not part of any subsystem. An instrument interprets them in the same way, regardless of the current path setting.

**Command and Query**

A SCPI command can be an Event command, Query command (a command that asks the analyzer for information), or both. The following are descriptions and examples of each form of command. GPIB Command Finder lists every SCPI command that is recognized by the analyzer, and its form.

<table>
<thead>
<tr>
<th>Form</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Event commands</strong> - cause an action to occur inside the analyzer.</td>
<td>:INITIATE:IMMEDIATE</td>
</tr>
<tr>
<td><strong>Query commands</strong> - query only; there is no associated analyzer state to set.</td>
<td>:SYSTem:ERRor?</td>
</tr>
<tr>
<td><strong>Command and query</strong> - set or query an analyzer setting. The query form appends a question mark (?) to the set form</td>
<td>:FORMat:DATA ! Command :FORMat:DATA? ! Query</td>
</tr>
</tbody>
</table>

**Multiple Commands**

You can send multiple commands within a single program message. By separating the commands with semicolons the current path does not change. The following examples show three methods to send two commands:

1. **Two program messages:**
   
   :SENSE:FREQUENCY:START 1e9;
   :SENSE:FREQUENCY:STOP 2e9

2. **One long message.** A colon follows the semicolon that separates the two commands causing the command parser to reset to the root of the command tree. As a result, the next command is only valid if it includes the entire keyword path from the root of the tree:
   
   :SENSE:FREQUENCY:START 1e9; :SENSE:FREQUENCY:STOP 2e9

3. **One short message.** The command parser keeps track of the position in the command tree. Therefore, you can simplify your program messages by including only the keyword at the same level in the command tree.
   
   SENSE:FREQUENCY:START 1e9;STOP 2e9

**Common Commands and SCPI Commands**

You can send Common commands and SCPI commands together in the same message. (For more information on these types of commands see GP-IB Fundamentals.) As in sending multiple SCPI commands, you must separate them with a semicolon.

Example of Common command and SCPI commands together

\*RST; SENSE:FREQUENCY:CENTER 5MHZ; SPAN 100KHZ

**Command Abbreviation**

Each command has a long form and an abbreviated short form. The syntax used in this Help system use uppercase characters to identify the short form of a particular keyword. The remainder of the keyword is lower case to complete the long form.
SENS - Short form
SENSE - Long form

Either the complete short form or complete long form must be used for each keyword. However, the keywords used to make a complete SCPI command can be a combination of short form and long form.

The following is unacceptable - The first three keywords use neither short or long form.

\texttt{SOURcE:POWer:ATTen:Auto on}

The following is acceptable - All keywords are either short form or long form.

\texttt{SOUR:POWer:ATT:AUTO on}

In addition, the analyzer accepts lowercase and uppercase characters as equivalent as shown in the following equivalent commands:

\texttt{source:POW:att:auto ON}
\texttt{Source:Pow:Att:Auto on}

Optional [Bracketed] Keywords

You can omit some keywords without changing the effect of the command. These optional, or default, keywords are used in many subsystems and are identified by brackets in syntax diagrams.

\textbf{Example of Optional Keywords}

The \texttt{SENSe} keyword is always optional. Therefore, both of the following commands are equivalent:

\texttt{:SENSe:BWID 1e3}
\texttt{:BWID 1e3}

The syntax in this Help system looks like this:

\texttt{[:SENSe]:BWID}

Vertical Bars | Pipes

Vertical bars, or "pipes", can be read as "or". They are sometimes used in syntax diagrams to separate alternative parameter options.

\textbf{Example of Vertical Bars:}

\texttt{SOURce:POWer:ATTenuation:AUTO <on|off>}

Either ON or OFF is a valid parameter option.

\textbf{MIN and MAX Parameters}

The special form parameters "MINimum" and "MAXimum" can be used with commands that specify single frequency (Hz) and time (seconds) as noted in the command documentation. \textbf{Note}: Also with these commands, kHZ, MHz, and GHz are accepted as suffixes/units.

The short form (min) and long form (minimum) of these two keywords are equivalent.

- \texttt{MAXimum} refers to the largest value that the function can currently be set to
- \texttt{MINimum} refers to the smallest value that the function can currently be set to.

\textbf{For example}, the following command sets the start frequency to the smallest value that is currently possible:

\texttt{SENS:FREQ:START MIN}

In addition, the max and min values can also be queried for these commands.

\textbf{For example}, the following command returns the smallest value that Start Frequency can currently be set to:

\texttt{SENS:FREQ:START? MIN}
An error will be returned if a numeric parameter is sent that exceeds the MAX and MIN values. 

**For example**, the following command will return an "Out of range" error message.

```
SENS:FREQ:START 1khz
```

---

**SCPI Errors**

**SCPI Errors**
- **-100 to -200 Command Errors**
- **-200 to -299 Execution Errors**
- **-300 to -399 SCPI Specified Device-Specific Errors**
- **-400 to -800 Query and System Errors**
- **100 to 200 FieldFox-specific Errors**

**See Also**

*How to Query the Error Queue*

---

**-100 to -200 Command Errors**

A command error indicates that the test set's GPIB parser has detected an IEEE 488.2 syntax error. When one of these errors is generated, the command error bit in the event status register is set.

- **-100 std_command**  Command - This event bit (Bit 5) indicates a syntax error, or a semantic error, or a GET command was entered, see IEEE 488.2, 11.5.1.1.4.
- **-101 std_invalidChar** Invalid character - Indicates a syntactic elements contains a character which is invalid for that type.
- **-102 std_syntax** Syntax - Indicates that an unrecognized command or data type was encountered. For example, a string was received when the device does not accept strings.
- **-103 std_invalidSeparator** Invalid separator - The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit.
- **-104 std_wrongParamType** Data type - The parser recognized a data element different than one allowed. For example, numeric or string data was expected but block data was encountered.
- **-105 std_GETNotAllowed** GET not allowed - Indicates a Group Execute Trigger was received within a program message. Correct the program so that the GET does not occur within the program code.
- **-108 std_tooManyParameters** Parameter not allowed - Indicates that more parameters were received than expected for the header. For example, *ESE common command only accepts one parameter, so *ESE 0,1 is not allowed.
- **-109 std_tooFewParameters** Missing parameter - Indicates that less parameters were received than required for the header. For example, *ESE requires one parameter,
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>*ESE is not allowed.</td>
</tr>
<tr>
<td>111</td>
<td>Command header - Indicates an error was detected in the header. This error is used when the device cannot detect the more specific errors -111 through -119.</td>
</tr>
<tr>
<td>112</td>
<td>Header separator - Indicates that a character that is not a legal header separator was encountered while parsing the header.</td>
</tr>
<tr>
<td>113</td>
<td>Program mnemonic too long - Indicates that the header contains more than twelve characters, see IEEE 488.2, 7.6.1.4.1.</td>
</tr>
<tr>
<td>114</td>
<td>Undefined header - Indicates the header is syntactically correct, but it is undefined for this specific device. For example, *XYZ is not defined for any device.</td>
</tr>
<tr>
<td>115</td>
<td>Header suffix out of range - Indicates the value of a header suffix attached to a program mnemonic makes the header invalid.</td>
</tr>
<tr>
<td>120</td>
<td>Numeric data - This error, as well as errors</td>
</tr>
<tr>
<td>121</td>
<td>Invalid character in number - Indicates an invalid character for the data type being parsed was encountered. For example, an alpha in a decimal numeric or a &quot;9&quot; in octal data.</td>
</tr>
<tr>
<td>123</td>
<td>Exponent too large - Indicates the magnitude of an exponent was greater than 32000, see IEEE 488.2, 7.7.2.4.1.</td>
</tr>
<tr>
<td>124</td>
<td>Too many digits - Indicates the mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros, see IEEE 488.2, 7.7.2.4.1.</td>
</tr>
<tr>
<td>128</td>
<td>Numeric data not allowed - Indicates that a legal numeric data element was received, but the device does not accept one in this position for the header.</td>
</tr>
<tr>
<td>130</td>
<td>Suffix - This error, as well as errors -131 through -139, are generated when parsing a suffix. This particular error message is used if the device cannot detect a more specific error.</td>
</tr>
<tr>
<td>131</td>
<td>Invalid suffix - Indicates the suffix does not follow the syntax described in IEEE 488.2, 7.7.3.1, or the suffix is inappropriate for this device.</td>
</tr>
<tr>
<td>134</td>
<td>Suffix too long - Indicates the suffix contain more than 12 characters, see IEEE 488.2, 7.7.3.4.</td>
</tr>
<tr>
<td>138</td>
<td>Suffix not allowed - Indicates that a suffix was encountered after a numeric element that does not allow suffixes.</td>
</tr>
<tr>
<td>140</td>
<td>Character data - This error, as well as errors</td>
</tr>
<tr>
<td>141</td>
<td>Invalid character data - Indicates that the character data element contains an invalid character or the particular element received is not valid for the header.</td>
</tr>
<tr>
<td>144</td>
<td>Character data too long - Indicates the character data element contains more than twelve characters, see IEEE 488.2, 7.7.1.4.</td>
</tr>
<tr>
<td>148</td>
<td>Character data not allowed - Indicates a legal character data element was encountered where prohibited by the device.</td>
</tr>
</tbody>
</table>
### -150 std_stringData
String data - This error, as well as errors

### -151 std_stringInvalid
Invalid string data - Indicates that a string data element was expected, but was invalid, see IEEE 488.2, 7.7.5.2. For example, an END message was received before the terminal quote character.

### -158 std_stringNotAllowed
String data not allowed - Indicates that a string data element was encountered but was not allowed by the device at this point in parsing.

### -160 std_blockData
Block data - This error, as well as errors -161 through -169, are generated when parsing a block data element. This particular error message is used if the device cannot detect a more specific error.

### -161 std_badBlock
Invalid block data - Indicates a block data element was expected, but was invalid, see IEEE 488.2, 7.7.6.2. For example, an END message was received before the end length was satisfied.

### -168 std_blockNotAllowed
Block data not allowed - Indicates a legal block data element was encountered, but not allowed by the device at this point in parsing.

### -170 std_expr
Expression - This error, as well as errors -171 through -179, are generated when parsing an expression data element. This particular error message is used if the device cannot detect a more specific error.

### -171 std_invalidExpression
Invalid expression - Indicates the expression data element was invalid, see IEEE 488.2, 7.7.7.2. For example, unmatched parentheses or an illegal character.

### -178 std_exprNotAllowed
Expression data not allowed - Indicates a legal expression data was encountered, but was not allowed by the device at this point in parsing.

### -180 std_macro
Macro - This error, as well as error -181 through -189, are generated when defining a macro or execution a macro. This particular error message is used if the device cannot detect a more specific error.

### -181 std_validOnlyInsideMacro
Invalid outside macro definition - Indicates that a macro parameter place holder was encountered outside of a macro definition.

### -183 std_invalidWithinMacro
Invalid inside macro definition - Indicates that the program message unit sequence, sent with a *DDT or a *DMC command, is syntactically invalid, see IEEE 488.2, 10.7.6.3.

### -184 std_macroParm
Macro parameter - Indicates that a command inside the macro definition had the wrong number or type of parameters.

### -200 to -299 Execution Errors
These errors are generated when something occurs that is incorrect in the current state of the instrument. These errors may be generated by a user action from either the remote or the manual user interface

### -200 std_execGen
Execution - This event bit (Bit 4) indicates a PROGRAM DATA element following a header was outside the legal input range or otherwise inconsistent with the device’s capabilities, see IEEE 488.2, 11.5.1.1.5.

### -201 std_invalidWhileInLocal
Invalid while in local

### -202 std_settingsLost
Settings lost due to rtl

### -203 std_commandProtected
Command protected - Indicates that a legal password-protected program command or query could not be executed because the command was
-210 std_trigger Trigger
disabled.

-211 std_triggerIgnored Trigger ignored

-212 std_armIgnored Arm ignored

-213 std_initIgnored Init ignored

-214 std_triggerDeadlock Trigger deadlock

-215 std_armDeadlock Arm deadlock

-220 std_parm Parameter - Indicates that a program data element related error occurred.

-221 std_settingsConflict Settings conflict - Indicates that a legal program data element was parsed but could not be executed due to the current device state.

-222 std_dataOutOfRange Data out of range - Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range defined by the devices.

-223 std_tooMuchData Too much data - Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.

-224 std_illegalParmValue Illegal parameter value - Indicates that the value selected was not part of the list of values given.

-225 std_noMemoryForOp Out of memory - The device has insufficient memory to perform the requested operation.

-226 std_listLength Lists not same length - Attempted to use LIST structure having individual LIST's of unequal lengths.

-230 std_dataCorruptOrStale Data corrupt or stale - Indicates invalid data, a new reading started but not completed since the last access.

-231 std_dataQuestionable Data questionable - Indicates that measurement accuracy is suspect.

-232 std_invalidFormat Invalid format

-233 std_invalidVersion Invalid version - Indicates that a legal program data element was parsed but could not be executed because the version of the data is incorrect to the device. For example, a not supported file version, a not supported instrument version.

-240 std_hardware Hardware - Indicates that a legal program command or query could not be executed because of a hardware problem in the device.

-241 std_hardwareMissing Hardware missing - Indicates that a legal program command or query could not be executed because of missing device hardware. For example, an option was not installed.

-250 std_massStorage Mass storage - Indicates that a mass storage error occurred. The device cannot detect the more specific errors described for errors -251 through -259.

-251 std_missingMassStorage Missing mass storage - Indicates that a legal program command or query
252 std_missingMedia Missing media - Indicates that a legal program command or query could not be executed because of missing media. For example, no disk.

253 std_corruptMedia Corrupt media - Indicates that a legal program command or query could not be executed because of corrupt media. For example, bad disk or wrong format.

254 std_mediaFull Media full- Indicates that a legal program command or query could not be executed because the media is full. For example, there is no room left on the disk.

255 std_directoryFull Directory full - Indicates that a legal program command or query could not be executed because the media directory was full.

256 std_fileNotFound File name not found - Indicates that a legal program command or query could not be executed because the file name was not found on the media.

257 std_fileName File name - Indicates that a legal program command or query could not be executed because the file name on the device media was in error. For example, an attempt was made to read or copy a nonexistent file.

258 std_mediaProtected Media protected - Indicates that a legal program command or query could not be executed because the media was protected. For example, the write-protect switch on a memory card was set.

260 std_expression Expression

261 std_math Math in expression

270 std_macroExecution Macro - Indicates that a macro related execution error occurred.

271 std_macroSyntax Macro syntax - Indicates that a syntactically legal macro program data sequence, according to IEEE 488.2, 10.7.2, could not be executed due to a syntax error within the macro definition.

272 std_macroExec Macro execution - Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition, see IEEE 488.2, 10.7.6.3.

273 std_badMacroName Illegal macro label - Indicates that the macro label was not accepted, it did not agree with the definition in IEEE 488.2, 10.7.3

274 std_macroPlaceholderMacro parameter - Indicates that the macro definition improperly used a macro parameter placeholder, see IEEE 4882, 10.7.3.

275 std_macroTooLong Macro definition too long - Indicates that a syntactically legal macro program data sequence could not be executed because the string of block contents were too long for the device to handle, IEEE 488.2, 10.7.6.1.

276 std_macroRecursion Macro recursion - Indicates that a syntactically legal macro program data sequence count not be executed because it would be recursive, see IEEE 488.2, 10.7.6.6.

277 std_cantRedefineMacro Macro redefinition not allowed - Indicates that redefining an existing macro label, see IEEE 488.2, 10.7.6.4.

278 std_macroNotFound Macro header not found - Indicates that a legal macro label in the
*GMS?, see IEEE 488.2, 10.13, could not be executed because the header was not previously defined.

-280 std_program Program
-281 std_cantCreateProgram Cannot create program
-282 std_illegalProgramName Illegal program name
-283 std_illegalVarName Illegal variable name
-284 std_programRunning Program currently running
-285 std_programSyntax Program syntax
-286 std_programRuntime Program runtime
-290 std_memoryUse Memory use
-291 std_execOutOfMemory Out of memory
-292 std_nameNotFound Referenced name does not exist
-293 std_nameAlreadyExists Referenced name already exists
-294 std_incompatibleType Incompatible type

-300 to -399 SCPI Specified Device-Specific Errors
A device-specific error indicates that the instrument has detected an error that occurred because some operations did not properly complete, possibly due to an abnormal hardware or firmware condition. For example, an attempt by the user to set an out of range value will generate a device specific error. When one of these errors is generated, the device specific error bit in the event status register is set.

-300 std_deviceSpecific Device specific - This event bit (Bit 3) indicates that a device operation did not properly complete due to some condition, such as overrange see IEEE 488.2, 11.5.1.1.6.
-310 std_system System
-311 std_memory Memory - Indicates some physical fault in the devices memory, such as a parity error.
-312 std_PUDmemoryLost PUD memory lost - Indicates protected user data saved by the *PUD command has been lost, see IEEE 488.2, 10.27.
-313 std_calMemoryLost Calibration memory lost - Indicates that nonvolatile calibration data used by the *CAL? command has been lost, see IEEE 488.2, 10.2.
-314 std_savRclMemoryLost Save/recall memory lost - Indicates that the nonvolatile data saved by the *SAV command has been lost, see IEEE 488.2, 10.33.
-315 std_configMemoryLost Configuration memory lost - Indicates that nonvolatile configuration data saved by the device has been lost.
-320 std_storageFault Storage fault - Indicates that the firmware detected a fault when using data storage. This is not an indication of physical damage or failure of any mass storage element.
-321 std_outOfMemory Out of memory - An internal operation needed more memory than was available
-330 std_selfTestFailed Self-test failed - Indicates a problem with the device that is not covered by
a specific error message. The device may require service.

-340 std_calFailed  Calibration failed - Indicates a problem during calibration of the device that is not covered by a specific error.

-350 std_queueOverflow  Queue overflow - Indicates that there is no room in the queue and an error occurred but was not recorded. This code is entered into the queue in lieu of the code that caused the error.

-360 std_comm  Communication - This is the generic communication error for devices that cannot detect the more specific errors described for error -361 through -363.

-361 std_parity  Parity in program message - Parity bit not correct when data received for example, on a serial port.

-362 std_framing  Framing in program message - A stop bit was not detected when data was received for example, on a serial port (for example, a baud rate mismatch).

-363 std_inputBufferOverrun  Input buffer overrun - Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing.

-400 to -800 Query and System Errors

A Query error is generated either when data in the instrument's GPIB output queue has been lost, or when an attempt is being made to read data from the output queue when no output is present or pending.

-400 std_queryGen  Query - This event bit (Bit 2) indicates that an attempt to read data from the Output Queues when no output is present or pending, to data in the Output Queue has been lost see IEEE488.2, 11.5.1.1.7.

-410 std_interrupted  Query INTERRUPTED - Indicates the test set has been interrupted by a new program message before it finishes sending a RESPONSE MESSAGE see IEEE 488.2, 6.3.2.3.

-420 std_unterminated  Query UNTERMINATED - Indicates an incomplete Query in the program see IEEE 488.2, 6.3.2.2.

-430 std_deadlocked  Query DEADLOCKED - Indicates that the Input Buffer and Output Queue are full see IEEE 488.2, 6.3.1.7.

-440 std_responseNotAllowed  Query UNTERMINATED after indefinite response - Indicates that a query was received in the same program message after a query requesting an indefinite response was executed see IEEE 488.2, 6.5.7.5.

-500 std_powerOn  Power on

-600 std_userRequest  User request

-700 std_requestControl  Request control

-800 std_operationComplete  Operation complete

Analyzer-Specific (Positive) SCPI Errors

-115 UnexpectedNumberOfParameters  The number of parameters received does not correspond to the number of parameters expected.
Last Modified:

4-Aug-2009  Cosmetic mods
Examples

FieldFox Programming Examples

- NA
- VVM
- VVS
- Calibration
- ECaI
- FOPS
- Guided Calibration
- Markers
- Memory Commands
- C# Example Program (Make a connection)
- Read Block Data using C#
- Transfer Image to PC

MATLAB Examples
- Binary Block File Transfer via VISA
- Acquire S21 Data via LAN as Socket (Port 5025)

Python Example
- Import SA Data as ASCII

VEE Examples
- Get NA Formatted Data as Real 32 BinBlock
- Get NA Formatted Data as ASCII Output
- SpecAn Get Data as ASCII Output
- Trigger Synch Sweep Complete

See Also
- How the FieldFox Error Queue Works
- Instrument Console (IC)
- SCPI Concepts and Tips

NA Mode Setup

This example shows how to:
- Create a simple NA Mode setup with 4-window configuration
- Set format, scale, frequency, IFBW, Averaging
- Set triggering to Single
- Read Data
- Clear Averaging
Note: In the following example:
"-p" indicates a user prompt.
"#" indicates a comment
This example can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. Learn more.

# Preset Instrument and Hold
*RST
# Change to NA Mode and wait until changed
INST:SEL "NA";*OPC?
# 4 window configuration
DISP:WIND:SPL D12_34
#change window 4 to R1 measurement
CALC:PAR4:DEF R1
#Select window 4
CALC:PAR4:SEL
#change window 4 format to SWR
CALC:FORMat SWR
# Autoscale window 1
DISP:WIND:TRAC1:Y:AUTO
# Set Center Freq
SENS:FREQ:CENT 2e9
# Set Freq Span
SENS:FREQ:SPAN 500e6
#set IFBW to 10 kHz
BWID:10e3
# Set averaging to 5 sweeps
AVER:COUNt 5
#Select window 1
CALC:PAR1:SEL
# Set to HOLD mode; wait
INIT:CONT 0;*OPC?
# Trigger FIVE measurements
# Wait between each
INIT:IMMediate;*OPC?
INIT:IMMediate;*OPC?
INIT:IMMediate;*OPC?
INIT:IMMediate;*OPC?
INIT:IMMediate;*OPC?
#Read formatted data from selected trace (1)
VVM Mode Example

The following example sets up a VVM measurement.

**Note:** In the following example:
- `-p` indicates a user prompt.
- `#` indicates a comment

This example can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. Learn more.

```
# Preset Instrument
SYST:PRESet;*OPC?
# Change to VVM Mode and wait until changed
INST:SEL "VVM";*OPC?
# Set to single sweep
INIT:CONT 0;*OPC?
# Setup the instrument for 1-port cable trimming
CALC:PAR:DEF S11
# Set Center Freq
SENS:FREQ:CENT 2e9
# Take a sweep to ensure you get a valid point on your new frequency.
INIT:IMM;*OPC?
# Attach the Reference cable
-p "Attach Reference Cable"
# Zero the display
SENS:CORR:ZERO:STAT ON;*OPC?
# Take another sweep to show the zero in action
INIT:IMM;*OPC?
# Prompt to attach a cable
-p "Attach Cable to Trim"
```
# Put into freerun mode
INIT:CONT ON

VVS Example

The following example sets up the VVS (variable voltage source).

**Note:** In the following example:

"-p" indicates a user prompt.

"#" indicates a comment

This example can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. [Learn more.](#)

The following example focuses on reliable switching (On and Off) of the VVS. The key is to make sure that the query reflects the current condition. This is important because the query result is NOT always updated.

The following is **pseudocode** as the Instrument Console has limited programming capability.

```
# This section ensures that the VVS is indeed ON after being tripped.
# Send VVS query
SYST:VVS?
# If "TRIPPED" is returned, then disable VVS
SYST:VVS:ENAB 0
# Wait 50 ms, then enable.
SYST:VVS:ENAB 1
# wait 50 ms, then repeat query.
SYST:VVS?
# If ON, then finished. Otherwise, send ENABle again, and then another query.

# This section ensures that the VVS is indeed OFF.
# Send VVS query
SYST:VVS?
# If "TRIPPED" is returned, then disable VVS
# If OFF, then finished. Otherwise...
SYST:VVS:ENAB 0
# Wait 50 ms, then repeat query.
SYST:VVS?
# Wait 50 ms, then repeat query.
# If OFF, then finished. Otherwise, send ENABle 1, then ENABle 0, and then another query.
SYST:VVS:ENAB 1
```
# Wait 50 ms.
SYST:VVS:ENAB 0
# wait 50 ms, then repeat query.
SYST:VVS?

Calibration Examples

The following examples show how to perform various calibrations in CAT and NA modes.

About Calibration Settings
A calibration session that is performed using the front-panel is completely separate from a calibration session that is performed programmatically.

Therefore, calibration settings that are made remotely (such as setting the cal kit and connectors) can NOT be observed from the front-panel user interface. Alternatively, calibration settings are that are made from the FieldFox front panel (user interface) can NOT be queried programmatically.

Guided Cal
- Guided Calibration (separate topic)
- ECal (separate topic)

Mechanical Cals
- 1-Port OSL
- 2- Port SOLR
- 2-Port QSOFT
- 2-Port Enhanced Response
- THRU Response

QuickCals
- 1-Port QuickCal
- 2-Port Non-Insertable QuickCal
- 2-Port Insertable QuickCal
- Enhanced Response QuickCal

Note: To Cal a VVM mode measurement, perform a Cal in CAT or NA mode, then switch to VVM mode.

See Also
- CAT Mode Commands
- NA Mode Commands

See All Programming Examples

Note: In the following examples:
"-p" indicates a user prompt.
Examples

"#" indicates a comment
Examples can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. Learn more.

1- Port OSL Cal on port 1
To perform a 1-port cal on port 2, replace all the '1' arguments with '2'.

```
# Select a Type N male, 50 ohm connector
# as the DUT connector to be attached to port 1.
# Be careful with the dashes in -M- for the following command.
# Some editors will change the character.
CORR:COLL:CONN 1, "Type N -M-,50"
# select the calkit to use.
CORR:COLL:CKIT:LABel 1,"85032B/E"
# Select 1-port cal (SOLT1) on port 1
CORR:COLL:METH:SOLT1 1
-p attach load to port 1
CORR:COLL:LOAD 1;*OPC?
-p attach short to port 1
CORR:COLL:SHOR 1;*OPC?
-p attach open to port 1
CORR:COLL:OPEN 1;*OPC?
# Finish and apply the cal
CORR:COLL:SAVE 0
```

2-Port SOLR Cal

```
# Setup 2-port SOLR Unknown Thru Cal between ports 1 and 2*
# Use the Type N male T kit on both ports
# Cal with either insertable or non-insertable connectors
# Do NOT use on N9912A
corr:coll:ckit:lab 1, "1250-3607"
corr:coll:ckit:lab 2, "1250-3607"
# Be careful with the dashes in -M- for the following command.
# Some editors will change the character.
corr:coll:conn 1,"Type N -M-,50"
corr:coll:conn 2,"Type N -M-,50"
CORR:COLL:METH:SOLR 1,2
-p Attach load to port 1
corr:coll:load 1;*OPC?
-p Attach short to port 1
corr:coll:shor 1;*OPC?
-p Attach open to port 1
corr:coll:open 1;*OPC?
-p Attach load to port 2
```
corr:coll:load 2;*OPC?
-p Attach short to port 2
corr:coll:shor 2;*OPC?
-p Attach open to port 2
corr:coll:open 2;*OPC?
-p Attach thru between ports 1 and 2
corr:coll:thru 1,2;*OPC?
corr:coll:save 0

### 2-Port QSOLT

# Setup 2-port QSOLT between port 1 and 2 using the T kit on port 1*
# OSL standards measured on port 2 - not available from the GUI
# For insertable DUTs ONLY
# Do NOT use on N9912A
corr:coll:ckit:lab 1, "1250-3607"
# Be careful with the dashes in -M- for the following command.
# Some editors will change the character.
corr:coll:conn 1,"Type N -M-,50"
corr:coll:conn 2,"Type N -F-,50"
corr:coll:meth:QSOL 1,2
-p Attach load to port 1
corr:coll:load 1;*OPC?
-p Attach short to port 1
corr:coll:shor 1;*OPC?
-p Attach open to port 1
corr:coll:open 1;*OPC?
-p Attach thru between ports 1 and 2
corr:coll:thru 1,2;*OPC?
corr:coll:save 0

### 2-Port Enhanced Response

# Enhanced Response requires an insertable DUT
# N9923A, calibrate measurements in either forward or reverse direction.
# N9912A, calibrate measurements in forward direction ONLY.
# This example is forward direction
# Choose the 85052D kit for port 1 and also port 2
corr:coll:ckit:lab 1, "85052D"
corr:coll:ckit:lab 2, "85052D"
# Choose connectors for port 1 and then port 2
# Be careful with the dashes in -M- for the following command.
# Some editors will change the character.
Examples

```
corr:coll:conn 1,"3.5 mm -M-,50"
corr:coll:conn 2,"3.5 mm -F-,50"
# Choose Enhanced Response forward
# For reverse measurement, use <2,1> and measure stds on port 2
corr:coll:meth:ERES 1,2
# Start measuring standards:
-p Attach short to port 1
corr:coll:shor 1;*OPC?
-p Attach open to port 1
corr:coll:open 1;*OPC?
-p Attach load to port 1
corr:coll:load 1;*OPC?
-p Connect ports 1 and 2 with Flush Thru
corr:coll:thru 1,2;*OPC?
# Saves the finished calibration
corr:coll:save 0
```

**THRU Response - Normalization**

```
# Correct transmission measurements
# N9912A must have Opt 110
SENS:CORR:COLL:METH:THRU 1,2
-p Attach thru now
SENS:CORR:COLL:THRU 1,2;*OPC?
# Finish and apply the cal
SENS:CORR:COLL:SAVE 0
```

**QuickCals**

1-Port QuickCal

```
CORR:COLL:METH:QCAL:CAL 1
# First step required to measure internal standards
# Port 1 must be left open
CORR:COLL:INT 1;*OPC?
-p (Optional) Attach load to port
CORR:COLL:LOAD 1;*OPC?
CORR:COLL:SAVE 0
```

2-Port Non-Insertable QuickCal*

```
# Setup full 2-port cal between port 1 and 2 using QuickCal.
# For a non-insertable DUT (both Type N -M-) performs SOLR
# Measure INT OPEN, SHORT on BOTH ports
# The load measurement steps are optional.
# Do NOT use on N9912A
```
# Be careful with the dashes in -M- for the following commands.
# Some editors will change the character.
corr:coll:conn 1,"Type N -M-,50"
corr:coll:conn 2,"Type N -M-,50"
corr:coll:meth:QCAL:CAL 1,2
-p Leave port 1 and port 2 open (no connection)
corr:coll:int 1;*OPC?
corr:coll:int 2;*OPC?
-p Attach a load to port 1
corr:coll:load 1;*OPC?
p- Attach a load to port 2
corr:coll:load 2;*OPC?
p- Connect ports 1 and 2 using any adapter/thru
corr:coll:thru 1,2;*OPC?
corr:coll:save 0

## Enhanced Response QuickCal

# Enhanced Response requires an Insertable DUT
# N9923A, calibrate measurements in either forward or reverse direction.
# N9912A, calibrate measurements in forward direction ONLY.
# This example is forward direction
# For reverse measurement, use <2,1> and measure stds on port 2
# Choose connectors for port 1 and then port 2
# Be careful with the dashes in -M- for the following command.
# Some editors will change the character.
corr:coll:conn 1,"3.5 mm -M-,50"
corr:coll:conn 2,"3.5 mm -F-,50"
CORR:COLL:METH:QCAL:ERES 1,2
# Step 1 - Ports 1 and 2 must be left open
-p Leave ports 1 and 2 OPEN
CORR:COLL:INT 1;*OPC?
# Step 2 - (Optional)
-p Attach load to port 1
CORR:COLL:LOAD 1;*OPC?
# Step 3 - Connect ports
-p connect port 1 and port 2
CORR:COLL:THRU 1,2;*OPC?
CORR:COLL:SAVE 0

ECal Calibration

ECal requires the use of the Guided Cal acquisition command:
[:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire
The following two ‘Guided’ commands are optional:
[:SENSe]:CORRection:COLLect:GUIDed:SCOunt
[:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMpt

Note: In the following example:
"-p" indicates a user prompt.
"#" indicates a comment
This example can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. Learn more.

Relevant ECal module commands:
- [:SENSe]:CORRection:COLLect:CKIT:LABel
- [:SENSe]:CORRection:COLLect:ECAL:AORient

The following program performs a 2-port SOLR calibration using an ECal module that is connected to a FieldFox.
A 2-port ECal may have 1 or 3 three steps depending on if 'simple ECal' is set and if the ECal is insertable (can connect to both test ports simultaneously). If one of those conditions is NOT true, then the cal will require 3 steps.

If you would like to use the 3 step cal process, ensure that CORR:COLL:ECAL:SIMP 0 has been set.

A 1-port ECal would require only 1 step.

If you would like to use 1 step simple cal ensure that CORR:COLL:ECAL:SIMP 1 has been set.

The :CORRection:COLLect:GUIDed:SCOunt? command is used to query the number of steps required.

The following example is for a 2-port SOLR cal that uses 3 steps.

```plaintext
# First setup a 2-port measurement between port 1 and 2
# Change the following line to your DUT/ECal module connector type and gender.
# Be careful with the dashes in -F- for the following command.
# Some editors will change the character.
corr:coll:conn 1,"Type N -F-,50"
corr:coll:conn 2,"Type N -F-,50"

# Change to your model ECal module
corr:coll:ckit:lab 1, "N4431A"
corr:coll:ckit:lab 2, "N4431A"
CORR:COLL:METH:SOLR 1,2

# ECAl requires the use of the Guided cal acquisition
# do a For/Next loop, query the number of steps (N)
# CORR:COLL:GUID:SCO?
# CORR:COLL:GUID:STEP:PROM? <step num> // query the prompt (optional, but recommended!)
# CORR:COLL:GUID:STEP:ACQ <step num> ;*OPC?
# Otherwise, measure all three stds for SOLR ECal w/o prompts
CORR:COLL:GUID:STEP:ACQ 1 ;*OPC?
CORR:COLL:GUID:STEP:ACQ 2 ;*OPC?
CORR:COLL:GUID:STEP:ACQ 3 ;*OPC?

# Finish
CORR:COLL:SAVE 0
```

**FOPS Measurement**

The following example sets up a VVM measurement.
**Guided Calibration**

The following C# example program performs a 1-port cal:

```csharp
// Performs a 1-port guided cal on the specified port with the specified connector
// for that port and kit for that port.
//
```
void Do1PortGuidedCal(int port, string connector, string kitLabel)
{
    Instrument.Write(string.Format("sens:corr:coll:conn {0},{1}", port, connector));
    Instrument.Write (string.Format("sens:corr:coll:ckit:lab {0},{1}",port,kitLabel));
    Instrument.Write (string.Format("sens:corr:coll:meth:SOLT1 {0}", port));

    int steps = ReadInt("SENS:CORR:COLL:GUID:SCOunt?");
    for(int s = 1; s <= steps; ++s)
    {
        PromptUser(steptext); // blocks until user confirms they've performed the requested action
        Instrument.ReadInt("*OPC?");
    }
    Console.Write("Saving...");
    Instrument.Write ("CORR:COLL:SAVE 0");
    Instrument.ReadInt("*OPC?");
    Console.WriteLine("Done ");
}
# Set Freq Span
SENS:FREQ:SPAN 500e6
# Setup the instrument to measure Insertion Loss
CALC:PAR:DEF S21
# Set to single trigger
INIT:CONT 0
# Take a single sweep at new freq and measurement
INIT:IMM;*OPC?
# Create reference marker
CALC:MARK1 NORM
# Move the marker to 1.75 GHz
CALC:MARK1:X 1.75e9
# Change to Delta Marker
CALC:MARK1 DELT
# Move the delta marker to 2.0 GHz
CALC:MARK1:X 2e9
# Take a sweep
INIT:IMM;*OPC?
# Read the Y axis values of the marker
# First value is mag, second is zero
CALC:MARK1:Y?

Memory Command Examples

These commands are used for memory storage and retrieval.
See also MMEM:DATA.

**Note:** If you attempt to save a filename that has already been saved at the specified memory location, the FieldFox displays the following error message:

Error -257, File name error; Storage Path "[INTERNAL]:\my_file" is not allowed.

`:MMEMory:DATA "<file_name>",#ABC

This command writes <data> into "<file_name>", where <data> is in 488.2 block format. The FieldFox expects to see waveform data as block data (binary files). The IEEE standard 488.2-1992 section 7.7.6 defines block data. The following example shows how to structure a SCPI command for downloading waveform data where #ABC represents the block data.

"<file_name>" The file name can be the short name, full file path, or NWFM format.
- Use the short name ("my_file"); the file will be stored in the default internal FieldFox directory:
  [INTERNAL]:\InternalSD\UserData.
• Use the file path: "[INTERNAL]:\my_data_folder\my_file"  
  (recommended)
  Note: You will need to create "my_data_folder", before saving your data.

• Other examples of storage location syntax:
  "[USB_DISK]:\my_data_folder\my_file"
  "[SDCARD]:\my_data_folder\my_file"

• Examples of deleting files. See also MMEM:DEL.

  Delete file from active drive/folder
  MMEM:DEL "MyOldFile.sta"

  'Delete file from USB
  MMEM:DEL "[USB_DISK]:\MyOldFile.sta"

  # This character indicates the beginning of the data block.

  A Number of decimal digits present in B

  B Decimal number specifying the number of data bytes to follow in C

  C Actual binary waveform data

Note: The following commands are not supported for non-volatile waveform memory (NVWFM).

:MMEM:CATalog? "<file_system>"
This query outputs a list of the files from the specified file system. The return data will be in the following form:
<mem_used>,<mem_free>{,"<file_listing>"}.

:MMEM:CDIRectory "[<directory_name>]", "[<directory_name>]"
:MMEM:CDIRectory?
This command changes the directory name for a file system. If no parameter is specified, the directory is set to the *RST value. At *RST, this value is set to the default user data directory. The query returns the full path of the default directory.

:MMEM:COPY "<file_name>"","<file_name>"
This command makes a duplicate of the requested file.

:MMEM:DELe "<file_name>"",<directory_name>
This command removes a file from the specified directory.

:MMEM:MDIRectory <directory_name>
This command creates a new directory where the <directory name> parameter specifies the name of the new directory.
Examples

:MMEMory:MOVE "<src_file>","<src_file_1>"
This command renames the src_file to src_file_1.

:MMEMory:RDIRectory <directory_name>
This command removes a directory where the <directory_name> parameter specifies the name of the directory to be removed. All files and directories under the specified directory are also removed.

Last Modified:
22sep2017 Added 10.00.

C# Example Program
The following C# example demonstrates how to send SCPI commands to the FieldFox using a TCP socket connection over a LAN connection.

- It is NOT necessary that you know C# to write a SCPI program. It is ONLY necessary that you understand basic SCPI syntax. You can add or replace the SCPI commands in this example program with your own.
- If you ARE familiar with C#, you can Download the project files here. (Internet connection required).

Note: You can also send single SCPI commands to the FieldFox using this free Instrument Console program.

Requirements
To connect to the FieldFox and run SCPI programs, you must first download and install the Visual C# Express software from: http://www.microsoft.com/express/download/

Once the program is installed, search the PC hard drive for csc.exe. This file could be in the C:\Windows\Microsoft.NET directory.

NOTE: If more than one folder contains csc.exe, use the folder with the latest revision.

Write the SCPI program
The following procedure uses the example filename MyProgram.cs. You can use any filename that you like.

1. Copy the text in the shaded are below into a Notepad file and name it MyProgram.cs.
2. Write your SCPI program between the //Start your program here and //End your program here lines. Several example lines are provided to demonstrate the syntax in which the SCPI commands must be contained. See the SCPI Command Reference and Program Examples for more information.

Run the SCPI Program
Record the dynamically-assigned IP address of the FieldFox.

1. Shut down the FieldFox.
2. Connect the FieldFox to the Internet using a LAN connection.
3. Power ON the FieldFox.
4. On the FieldFox, press System, then System Configuration, then LAN.
5. Record the Current IP Address
6. Compile your program by executing csc.exe MyProgram.cs. This creates a file named MyProgram.exe in the same directory as csc.exe.
7. Run your SCPI program by executing MyProgram.exe <FieldFox IP Address>. For example: MyProgram.exe 192.121.1.101

To make this process more convenient:
The following steps show how to create shortcuts on your PC desktop to compile and run MyProgram.exe.

1. Using Windows Explorer, navigate to the folder that contains csc.exe. NOTE: If more than one folder contains csc.exe, use the folder with the latest revision.
2. Right-click csc.exe then click Create Shortcut.
3. Drag the shortcut file to the PC desktop.
4. Right-click on the desktop shortcut, then click Properties.
5. Append a space, the full path, and filename to the end of the "Target" as in the following image. This example shows MyProgram.cs is saved to the C:\ folder.

<table>
<thead>
<tr>
<th>Target type: Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target location: v3.5</td>
</tr>
<tr>
<td>Target:</td>
</tr>
<tr>
<td>\NET\Framework\v3.5\csc.exe  c:\MyProgram.cs</td>
</tr>
</tbody>
</table>

6. After performing a compile, perform the same ‘shortcut’ procedure for MyProgram.exe except, instead of appending the path and filename, append the IP address of the FieldFox.

<table>
<thead>
<tr>
<th>Target type: Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target location: C:\</td>
</tr>
<tr>
<td>Target:</td>
</tr>
<tr>
<td>C:\MyProgram.exe 123.456.1.789</td>
</tr>
</tbody>
</table>

Copy the text in the following shaded area to a Notepad file.

```csharp
using System;
using System.Collections.Generic;
using System.Text;
using System.Net.Sockets;
using System.IO;
namespace Network.Connect
{
    class Program
    {
        static TelnetConnection tc;
        static int Main(string[] args)
        {
            // defaultHostName is host name to use if one is not specified on the command line.
            string defaultHostName = "192.168.1.1";
```
string hostName = defaultHostName;
if( args.Length == 1 )
{
    // If command line contains a '?' character, interpret
    this as help.
    if( args[0].Contains("?"))
    {
        Console.WriteLine("Usage: N9912A_CS_Example.exe
<hostName>

" +
        "Where optional hostName is an ip address or host
name.
" +
        "If no hostName is supplied, the default
("+defaultHostName++) is used.
" +
        "e.g. N9912A_CS_Example.exe 10.10.1.1
nor
" +
        "N9912A_CS_Example.exe A-N9912A-22762");
        return 0; // exit.
    }
    // Record hostname passed in on command line.
    hostName = args[0];
}
try
{
    tc = new TelnetConnection();
    tc.ReadTimeout = 10000; // 10 sec
    // open socket on hostName, which can be an IP address, or
    use host name (e.g. "A-N9912A-22762") used in lieu of IP
    address
    tc.Open(hostName);
    if( tc.IsOpen )
    {
        //Start your program here
        Write("SYST:PRES:*OPC?");
        Write("*IDN?");
        Write("SENS:FREQ:STAR?");
        Write("SENS:FREQ:STAR 3e9");
        Write("SENS:FREQ:STAR?");
        Write("SYST:ERR?");
        Write("SYST:HELP:HEAD?");
        //End your program here
        tc.Dispose();
        Console.WriteLine("Press any key to exit.");
        Console.ReadKey(); // continue after reading a key
from the keyboard.
else
{
    Console.WriteLine("Error opening " + hostName);
    return -1;
}
//FieldFox Programming Guide 5
} catch(Exception e)
{
    Console.WriteLine(e.ToString());
    return -1;
}
// exit normally.
return 0;
} /// <summary>
/// Write a SCPI command to the telnet connection.
/// If the command has a '?', then read back the response and
print
/// it to the Console.
/// </summary>
/// <remarks>
/// Note the '?' detection is naive, as a ? could occur in the
middle
/// of a SCPI string argument, and not actually signify a SCPI
query.
/// </remarks>
/// <param name="s"></param>
static void Write(string s)
{
    Console.WriteLine(s);
    tc.WriteLine(s);
    if (s.IndexOf('?') >= 0)
        Read();
}
/// <summary>
/// Read the telnet connection for a response, and print the
response to the
/// Console.
/// </summary>
static void Read()
Example:

```csharp
Console.WriteLine(tc.Read());
}

#region TelnetConnection - no need to edit
/// <summary>
/// Telnet Connection on port 5025 to an instrument
/// </summary>
public class TelnetConnection : IDisposable
{
    TcpClient m_Client;
    NetworkStream m_Stream;
    bool m_IsOpen = false;
    string m_Hostname;
    int m_ReadTimeout = 1000; // ms
    public delegate void ConnectionDelegate();
    public event ConnectionDelegate Opened;
    public event ConnectionDelegate Closed;
    public bool IsOpen { get { return m_IsOpen; } }
    public TelnetConnection() { }
    public TelnetConnection(bool open) : this("localhost", true) { }
    public TelnetConnection(string host, bool open)
    {
        if (open)
            Open(host);
    }
    void CheckOpen()
    {
        if (!IsOpen)
            throw new Exception("Connection not open.");
    }
    public string Hostname
    {
        get { return m_Hostname; }
    }
    public int ReadTimeout
    {
        set { m_ReadTimeout = value; if (IsOpen) m_Stream.ReadTimeout = value; }
        get { return m_ReadTimeout; }
    }
    public void Write(string str)
    { }
}
CheckOpen();
byte[] bytes = System.Text.ASCIIEncoding.ASCII.GetBytes(str);
m_Stream.Write(bytes, 0, bytes.Length);
m_Stream.Flush();
}
public void WriteLine(string str)
{
    CheckOpen();
    byte[] bytes = System.Text.ASCIIEncoding.ASCII.GetBytes(str);
m_Stream.Write(bytes, 0, bytes.Length);
    WriteTerminator();
}
void WriteTerminator()
{
    byte[] bytes =
System.Text.ASCIIEncoding.ASCII.GetBytes("\r\n\0");
m_Stream.Write(bytes, 0, bytes.Length);
m_Stream.Flush();
}
public string Read()
{
    CheckOpen();
    return System.Text.ASCIIEncoding.ASCII.GetString(ReadBytes());
}
/// <summary>
/// Reads bytes from the socket and returns them as a byte[].
/// </summary>
/// <returns></returns>
public byte[] ReadBytes()
{
    int i = m_Stream.ReadByte();
    byte b = (byte)i;
    int bytesToRead = 0;
    var bytes = new List<byte>();
    if ((char)b == '#')
    {
        bytesToRead = ReadLengthHeader();
        if (bytesToRead > 0)
        {
            i = m_Stream.ReadByte();
            if (((char)i != '\n') // discard carriage return after
length header.
    bytes.Add((byte)i);
}

if (bytesToRead == 0)
{
    while (i != -1 && b != (byte)'
')
    {
        bytes.Add(b);
        i = m_Stream.ReadByte();
        b = (byte)i;
    }
}
else
{
    int bytesRead = 0;
    while (bytesRead < bytesToRead && i != -1)
    {
        i = m_Stream.ReadByte();
        if (i != -1)
        {
            bytesRead++;
            // record all bytes except \n if it is the last char.
            if (bytesRead < bytesToRead || (char)i != '
')
                bytes.Add((byte)i);
        }
    }
    return bytes.ToArray();
}

int ReadLengthHeader()
{
    int numDigits = Convert.ToInt32(new string(new char[] {
        (char)m_Stream.ReadByte() }));
    string bytes = "";
    for (int i = 0; i < numDigits; ++i)
    {
        bytes = bytes + (char)m_Stream.ReadByte();
        return Convert.ToInt32(bytes);
    }
}

public void Open(string hostname)
{
if (IsOpen)
    Close();

m_Hostname = hostname;

m_Client = new TcpClient(hostname, 5025);

m_Stream = m_Client.GetStream();

m_Stream.ReadTimeout = ReadTimeout;

m_IsOpen = true;

if (Opened != null)
    Opened();

public void Close()
{
    if (!m_IsOpen)
        //FieldFox Programming Guide 7
        return;

    m_Stream.Close();
    m_Client.Close();
    m_IsOpen = false;

    if (Closed != null)
        Closed();

}

public void Dispose()
{
    Close();
}

#region

#endregion

---

Read Block Data using Csharp

The following example program illustrates how to parse block data using C#.

/// <summary>
/// Generates a IEEE block header for the specified size.
/// </summary>
/// <remarks>
/// The block header is of the form #[digit indicating number of digits to follow][length]
/// e.g. 201 bytes -> "#3201"
/// 9999 bytes -> "#49999"
/// 0 bytes -> "#10"
/// </remarks>
/// <summary>
/// Parses a partially digested IEEE block length header, and returns
/// the specified byte length.
/// </summary>
/// <param name="numDigits">Number of digits to read from the stream that make up the
/// length in bytes.</param>
/// <returns>The length of the block.</returns>
int ReadLengthHeader(int numDigits)
{
    string bytes = string.Empty;
    for (int i = 0; i < numDigits; ++i)
    {
        bytes = bytes + (char)Stream.ReadByte();
    }
    return Convert.ToInt32(bytes);
}

---

Transfer Image to PC

This example shows how to transfer an image (screenshot) on the FieldFox to a remote PC.

**Note:** In the following example:
- "#" indicates a comment
- This example can be copied into a text editor, saved as a *.txt file, and run using the free Keysight Instrument Console program. [Learn more.](#)

```
# Store screen to my.png into the current directory on the FieldFox
```
```matlab
# The default directory is the userdata directory on the instrument.
MMEM:STOR:IMAG "my.png"
# Transfers the contents of my.png as a BINBLOCK
# The file data that is returned by the 2nd command depends on the
programming environment.
# Environments like VEE, Matlab, C/VISA, etc. all deal with BINBLOCK
transfers in their own way.
MMEM:DATA? "my.png"
# Optionally delete of file from instrument's local storage
MMEM:DEL "my.png"
```

MATLAB

MATLAB - Binary Block File Transfer Via VISA

This is a MATLAB example that enables you to control the FieldFox family of Combination Analyzers via a
VISA resource string.
- The program first clears the error queue and all status registers via the "*CLS" command.
- The *IDN? identification query is then asserted and the resultant string is read.
- The application stores a PNG file,'Test_Image.PNG',to the internal memory of the targeted FieldFox
(FF) analyzer.
- Next the stored PNG image file,'Test_Image.PNG', is transferred from the FieldFox to the controlling PC
via the MMEM:DATA? query. This transfer is by default always an IEEE-754 binary bin-block transfer.
- The .PNG file save on the controlling PC is stored as 'C:\Temp\TransferedTestImage.png'
- Lastly, the system error queue is checked at conclusion of the application. If no errors were generated
the response to the "SYST:ERR?" then the query will still read "+0, "No Error"".

Note:  In the following example:
"%;" or "%%" indicates a comment
"
";" indicates the end of a comment

```matlab
% Sample MATLAB program for the Keysight Technologies FieldFox (FF) handheld
combination analyzers.
The sample program connects to a FF Family handheld combination analyzer
thru a VISA resource string.
The program first clears the error queue and all status registers via the
"*CLS" command. The *IDN? identification query is then asserted and the
resultant string is read.
The application stores a PNG file,'Test_Image.PNG',to the internal memory
of the targeted
FieldFox (FF) analyzer.
```

45
Next the stored PNG image file, 'Test_Image.PNG', is transferred from the 
FF to the controlling PC via the MMEM:DATA? query. This transfer is by default always an IEEE-754 
binary bin-block transfer.

The .PNG file save on the controlling PC is stored as 
'C:\Temp\TransferedTestImage.png'

As a wrap up the system error queue is checked at conclusion of the 
application. If no errors were generated the response to the "SYST:ERR?" 
query will still be "+0, "No Error"".

{%
%Remove all interfaces to instrument

instrreset
% find all previously created objects
oldobjs = instrfind;
% If there are any existing objects
if (~isempty(oldobjs))
    % close the connection to the instrument
    fclose(oldobjs);
    % and free up the object resources
    delete(oldobjs);
end

% Remove the object list from the workspace.
clear oldobjs;
%
{Define FieldFox (FF) interface, this is the VISA resource string. Replace this VISA
resource string with your controlling PC's FieldFox VISA resource string as appropriate.
For this applicaiton the 'agilent' I/o libraries are utilized.
%
fieldFox = visa('agilent', 'TCPIP0::156.140.159.126::inst0::INSTR');
% Buffer size must precede open command
set(fieldFox, 'InputBufferSize', 640000);
set(fieldFox, 'OutputBufferSize', 640000);
% Open session to fieldFox based on VISA resource string
fopen(fieldFox);
% Clear the event status registers and all errors which may be in the 
FieldFox's queue.
fprintf(fieldFox, '*CLS');
% Check to ensure the error queue is clear. Response is "+0, No Error"
fprintf(fieldFox, 'SYST:ERR?');
[errIdentifyStart,~] = fscanf(fieldFox, '%c');


% Initial error check results: ', errIdentifyStart
fprintf(fieldFox, '*IDN?');
[idn,~] = fscanf(fieldFox, '%c');

% Instrument identified as: ', idn
% Set the FF mass storage to the internal drive
fprintf(fieldFox, 'MMEM:CDIR "[INTERNAL]:"');

% Binary efforts here
% First store an image to the local FF memory.
fprintf(fieldFox, 'MMEM:STOR:IMAG "TestImage.png"');

% Query image via MMEM:DATA? 'yourFileNameHere.mimeExtensionType'
fprintf(fieldFox, 'MMEM:DATA? "TestImage.png"');

% Dump return bits to a variable 'screenPNG' via a MATLAB binblockread call.
% MATLAB binblockread supports five 8-bit bin block read types:
%   uchar, schar, int8, unit8, char.
% Of these uint8, uchar, char all functioned without corrupting the binary bits,
% i.e., the resultant file transfer preserved the data integrity of the original
% file without corruption.
screenPNG = binblockread(fieldFox,'uint8'); fread(fieldFox,1);
% Write bits to file as PNG file save
% From MATLAB help (in command window 'help fid' to view details)
% 'FID = fopen(FILENAME) opens the file FILENAME for read access'.
% FILENAME is the name of the file to be opened. Thus, in this case open
% C:\Temp\TransferedTestImage.png
% The 'w' indicates 'open file for writing; discard existing contents'
fid = fopen('C:\Temp\TransferedTestImage.png','w');
fwrite(fid,screenPNG,'uint8');
fclose(fid);
% As a last step query the fieldFox error queue and ensure no errors have occurred since initiation and completion of the program
fprintf(fieldFox, 'SYST:ERR?');
[errIdentifyStop,~] = fscanf(fieldFox, '%c');

% Close session connection
fclose(fieldFox);
delete(fieldFox);
clear fieldFox;

% Import the saved image into MATLAB workspace
importedImage = imread('C:\Temp\TransferedTestImage.png')

image(importedImage)

['Initial error check results: ', errIdentifyStart]
['Instrument identified as: ', idn]
['Final error check results: ', errIdentifyStop]

MATLAB - Binary Block File Transfer Via LAN as Socket at Port 5025

This is a MATLAB example that enables you to control the FieldFox family of Combination Analyzers. This example does not rely the VISA libraries or a VISA connection. Connection to the targeted analyzer is via TCP/IP and sockets at port number 5025, as supported by the Keysight Technologies FieldFox handheld combination analyzers.

- The sample program sets the Keysight Technologies FieldFox handheld analyzer to "NA" (network analyzer) mode.
- The DUT is a 177MHz wideband band-pass filter.
- The application then acquires S21 transmission data and frequency stimulus data via binary bin-block data transfers.
- The resultant data is plotted in the MatLab GUI as a logMag versus frequency X-Y plot.

**Note:** In the following example:

"%" indicates a comment

```matlab
% Instantiate connection to Keysight FieldFox via LAN as Socket at Port 5025
% Alter the TCPIP address to match your targeted FieldFox IP address.
fieldFox = tcpip('156.140.155.106',5025);
% Set input and output buffer default sizes
set(fieldFox, 'InputBufferSize', 8096);
set(fieldFox, 'OutputBufferSize', 8069);
% Default binary data read is BigEndian resulting in corrupt data.
% Modify return of binary data from default BigEndian to LittleEndian
% via MathWorks SET command
set(fieldFox,'ByteOrder', 'littleEndian')
% Open session to fieldFox at address / port as noted above.
fopen(fieldFox);
% 'Hello World' equivalent, i.e. Identification Query String
fprintf(fieldFox,'*IDN?\n');
myId = fscanf(fieldFox,'%c')
% Clear the status registers and all potential error indications within the
% error queue prior to starting applications. Also, check the error queue
```
via
% 'SYST:ERR?' error query and ensure the error indication is '0, "No Error"

fprintf(fieldFox,'*CLS\n');
fprintf(fieldFox,'SYST:ERR?\n');
initErrCheck = fscanf(fieldFox,'%c')

% Set Instrument and various other important items
% Instrument mode to Network Analyzer
fprintf(fieldFox,'INST:SEL "NA"')
% Trigger mode to continuous off
fprintf(fieldFox,'INIT:CONT 0\n')
% Set start and stop frequencies. DUT is a wideband 177MHz bandpass filter (BPF).
fprintf(fieldFox,'FREQ:STAR 60E6;STOP 300E6\n')
% Set number of trace points
fprintf(fieldFox,'SWE:POIN 101\n')
% Trace 1 to measurement of S21 and select that measurement as active
fprintf(fieldFox,'CALC:PAR1:DEF S21;SEL\n')
% Hold off for operation complete to ensure settings
fprintf(fieldFox,'*OPC?\n')
done = fscanf(fieldFox,'%1d')

% Trigger single sweep with hold off via *OPC? Operation Complete Query.
% For long sweeps times there may be a TCPIP hold off or time out setting that
% must be increased.
fprintf(fieldFox,'INIT;*OPC?\n')
trigComplete = fscanf(fieldFox,'%1d')

%Query FORMATTED data from fieldFox
% Set data format to real-32 bin block transfer
fprintf(fieldFox, 'FORM:DATA REAL,32\n')
fprintf(fieldFox,'CALC:DATA:FDATA?\n')
myBinData = binblockread(fieldFox,'float')
% There will be a line feed not read, i.e. hanging. Read it to clear buffer.
% If you do not read the hanging line feed a -410, "Query Interrupted % Error" will occur
hangLineFeed = fread(fieldFox,1)

%Query of x-axis stimulus
% Set data format to real-64 bin block transfer. Real 64 bit to ensure
% Hz resolution in GHz capable analyzers.
fprintf(fieldFox, 'FORM:DATA REAL,64\n')
fprintf(fieldFox,'SENS:FREQ:DATA?\n')
myBinStimulusData = binblockread(fieldFox,'double')
  % There will be a line feed not read, i.e. hanging. Read it to clear buffer.
  hangLineFeed = fread(fieldFox,1)
  % Within the MatLab GUI display data and stimulus numbers and plot same
  display myBinData
  display myBinStimulusData

  % MatLab plot related commands and efforts:

  % Convert FieldFox returned frequency data to units of MHz
  myStimulusDataMHz = myBinStimulusData/1E6

  clear title xlabel ylabel

  plot(myStimulusDataMHz, myBinData)
  title('S21 : 177MHz Band Pass Filter')
  xlabel('Frequency (MHz)')
  ylabel ('Log Mag (dB)')

% Check Error Queue. A "*CLS" was asserted at the beginning of the
% application. This will clear the entire error queue. Upon completion of
% the application the error queue is queried a final time. If the
% application is written correctly and there are no hardware failures the
% final error query check via 'SYST:ERR?' should return '0, "No Error"
else
  % the application is in error.
  fprintf(fieldFox, 'SYST:ERR?')
  finalErrCheck = fscanf(fieldFox, '%c')
%Close session to instrument prior to completion
fclose(fieldFox);

---

**Python Example**

This is a python example that enables you to control the FieldFox family of Combination Analyzers to import SA data as ASCII.

- Imports the pyvisa libraries and operating system dependent functionality;
- Establishes a visa resource manager;
- Opens a connection to the FieldFox based on the instrument's VISA address as acquired via Keysight Connection Expert;
- Sets the visa time out (increasing the timeout as compared to the default).
- Clears the event status register and thus clears the error queue;
- Defines an error check function and checks the system error queue;
- Presets the FieldFox unit; performs a *IDN?, sets the analyzer to Spectrum Analyzer mode,
- Then queries the number of points, start frequency and stop frequency.
- Executes a synchronized single sweep.
- Queries the spectrum analyzer trace data, builds a linear array to compute the stimulus array, and
- Plots the stimulus - response data as an X-Y trace.

**Note:** In the following example:

"#" indicates a comment

```
# -*- coding: utf-8 -*-
# Python for Test and Measurement
# Requires VISA installed on controlling PC, 'http://pyvisa.sourceforge.net/pyvisa/'
# Keysight IO Libraries 18.1.22x 32-Bit Keysight VISA (as primary)
# Anaconda Python 4.4.0 32 bit
# pyvisa 3.6.x
# Keysight N9952A 50GHz FieldFox Handheld portable combination analyzer
# running A.10.17 application code
#
## Copyright © 2018 Keysight Technologies Inc. All rights reserved.
##
## You have a royalty-free right to use, modify, reproduce and distribute this
## example / files (and/or any modified version) in any way you find useful, provided
## that you agree that Keysight has no warranty, obligations or liability for any
## Sample Application / Files.
##
## Example Description:

# A python sample program utilizing pyvisa to connect and control a Keysight FieldFox
# Family Combination Analyzer.
#
# The application performs the following:

# Imports the pyvisa libraries and operating system dependent functionality;
# Establishes a visa resource manager;
# Opens a connection to the FieldFox based on the instrument's VISA address as
```
acquired via Keysight Connection Expert
# Sets the visa time out (increasing the timeout as compared to the default).
# Clears the event status register and thus clears the error queue;
# Defines an error check function and checks the system error queue;
# Presets the FieldFox unit; performs a *IDN?, sets the analyzer to Spectrum Analyzer mode,
# then queries the number of points, start frequency and stop frequency.
# Executes a synchronized single sweep.
# Queries the spectrum analyzer trace data, builds a linear array to compute the stimulus array,
# and plots the stimulus - response data as an X-Y trace.
#
# Import the visa libraries
import visa
import os
# The numpy is imported as it is helpful for a linear ramp creation for the stimulus array
import numpy as npStimulusArray
# import module for plotting
import matplotlib.pyplot as stimulusResponsePlot
# A variable to control various events and testing during development.
# by uncommenting the #debug True line, debug will occur, for efficiency, during development.
debug = False
#debug = True
print "Debug flag set to " + str(debug)

# Set variables for ease of change - assumes 'debug is true.
# If debug is set to false then Spectrum Analyzer preset defaults for
# start frequency, stop frequency and number of points are utilized.
numPoints = 21
startFreq = 1.28579E9
stopFreq = 2.28579E9

# Open a VISA resource manager pointing to the installation folder for the Keysight Visa libraries.
rm = visa.ResourceManager('C:\Program Files (x86)\IVI Foundation\VISA\WindNT\agvisa\agbin\visa32.dll')

# Based on the resource manager, open a session to a specific VISA resource string as provided via
# Keysight Connection Expert
# ALTER LINE BELOW - Updated VISA resource string to match your specific configuration
myFieldFox = rm.open_resource("TCPIP0::156.140.157.162::inst0::INSTR")
# Set Timeout - 10 seconds
myFieldFox.timeout = 10000

# Clear the event status registers and empty the error queue
myFieldFox.write("*CLS")

# Query identification string *IDN?
myFieldFox.write("*IDN?")

print (myFieldFox.read())

# Define Error Check Function
def Errcheck():
    myError = []
    ErrorList = myFieldFox.query("SYST:ERR?").split(', ')
    Error = ErrorList[0]
    if int(Error) == 0:
        print ("+0, No Error!")
    else:
        while int(Error) != 0:
            print ("Error #: " + ErrorList[0])
            print ("Error Description: " + ErrorList[1])
            myError.append(ErrorList[0])
            myError.append(ErrorList[1])
            ErrorList = myFieldFox.query("SYST:ERR?").split(', ')
            Error = ErrorList[0]
    myError = list(myError)
    return myError

# Call and print error check results
print (Errcheck())

# Preset the FieldFox and wait for operation complete via the *OPC?, i.e.
# the operation complete query.
myFieldFox.write("SYST:PRES;*OPC?")

print "Preset complete, *OPC? returned : " + myFieldFox.read()

# Set mode to Spectrum Analyzer and wait for operation complete via the *OPC?, i.e.
# the operation complete query.
myFieldFox.write("INST:SEL 'SA';*OPC?")

myFieldFox.read()

# If debug is true then user setting of start frequency, stop frequency and number of points
if debug:
    myFieldFox.write("SENS:SWE:POIN " + str(numPoints))
    myFieldFox.write("SENS:FREQ:START " + str(startFreq))
    myFieldFox.write("SENS:FREQ:STOP " + str(stopFreq))
# Determine, i.e. query, number of points in trace for ASCII transfer - query
myFieldFox.write("SENS:SWE:POIN?"")
numPoints = myFieldFox.read()
print "Number of trace points " + numPoints

# Determine, i.e. query, start and stop frequencies, i.e. stimulus begin and end points
myFieldFox.write("SENS:FREQ:START?"")
startFreq = myFieldFox.read()
myFieldFox.write("SENS:FREQ:STOP?"
stopFreq = myFieldFox.read()
print "FieldFox start frequency = " + startFreq + " stop frequency = " + stopFreq

# Set trigger mode to hold for trigger synchronization
myFieldFox.write("INIT:CONT OFF;*OPC?"")
myFieldFox.read()

# Use of Python numpy import to compute linear step size of stimulus array
# based on query of the start frequency - stop frequency and number of points.
# 'Other' analyzers support a SCPI "SENSe:X?" query which will provide the stimulus
# array as a SCPI query.
stimulusArray = npStimulusArray.linspace(float(startFreq),float(stopFreq),int(numPoints))
print stimulusArray

# Assert a single trigger and wait for trigger complete via *OPC? output of a 1
myFieldFox.write("INIT:IMM;*OPC?"")
print "Single Trigger complete, *OPC? returned : " + myFieldFox.read()

# Query the FieldFox response data
myFieldFox.write("TRACE:DATA?"
ff_SA_Trace_Data = myFieldFox.read()
print ff_SA_Trace_Data # This is one long comma separated string list of values.

# Use split to turn long string to an array of values
ff_SA_Trace_Data_Array = ff_SA_Trace_Data.split("",""

# Now plot the x - y data
maxResponseVal = max(ff_SA_Trace_Data_Array)
minResponseVal = min(ff_SA_Trace_Data_Array)

if debug:
print "Max value = " + maxResponseVal + " Min Value = " + minResponseVal

stimulusResponsePlot.title ("Keysight FieldFox Spectrum Trace Data via Python - PyVisa - SCPI")
stimulusResponsePlot.xlabel("Frequency")
stimulusResponsePlot.ylabel("Amplitude (dBm)"
stimulusResponsePlot.plot(stimulusArray,ff_SA_Trace_Data_Array)
stimulusResponsePlot.autoscale(True, True, True)
stimulusResponsePlot.show()
# Return the FieldFox back to free run trigger mode
myFieldFox.write("INIT:CONT ON")
# Send a corrupt SCPI command end of application as a debug test
if debug:
    myFieldFox.write("INIT:CONT OOOOOOOOOO")
# Call the ErrCheck function and ensure no errors occurred between start of program
# (first Errcheck() call and end of program (last Errcheck() call.
print (Errcheck())
# On exit clean a few items up.
myFieldFox.clear()
myFieldFox.close()

VEE

Get Formatted Data as Real 32 Binary Block
The following VEE example demonstrates how to configure measurement settings, then read formatted Real 32 bit Binary Block data.
Download N9912A_NA_Get_Fdata_as_Real32BinBlock_Output_PROGRAM.VEE
Get Formatted Data as ASCII Output

The following VEE example demonstrates how to configure NA measurement settings, then read formatted ASCII data from the FieldFox.

Download **N9912A_NA_Get_Fdata_as_ASCII_Output_PROGRAM.VEE**

![Diagram of VEE example](image)

Last Modified:

24-Aug-2011  New topic

Trigger Synch Sweep Complete

The following VEE example demonstrates how to configure measurement settings, set trigger to single, then notify when a sweep is complete.

Download **FieldFox_Trigger_Synchronization_Sweep_Complete_Program.VEE**
SpecAn Get Data as ASCII Output

The following VEE example demonstrates how to configure SA measurement settings, then read formatted ASCII data from the FieldFox.

Download [N9912A_SpecAn_Get_Data_as_ASCII_Output_PROGRAM.VEE](#)
Last Modified:

19-Dec-2018 New command
Commands by Mode

CAT Mode Commands

In this topic:

- Set and select Traces
- Sweep
- Averaging
- Display
- Limit Lines
- Markers
- Math
- DTF
- DTF Cable Correction
- TDR
- Save/Recall
- Send/Read Data
- Calibration

See Also
- Commands that are Common to All Modes
- Correction Methods Explained
- New Programming Commands
- Status Registers

Set and select Traces

<table>
<thead>
<tr>
<th>Description</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set and read number of traces</td>
<td>\texttt{CALCulate:PARameter:COUNT}</td>
</tr>
<tr>
<td>Change parameter</td>
<td>\texttt{CALCulate:PARameter:DEFine}</td>
</tr>
<tr>
<td>Select trace</td>
<td>\texttt{CALCulate:PARameter:SELect}</td>
</tr>
</tbody>
</table>

Sweep Settings

- Set center freq: \texttt{[:SENSe]:FREQuency:CENTer}
- Set freq span: \texttt{[:SENSe]:FREQuency:SPAN}
- Set start freq: \texttt{[:SENSe]:FREQuency:START}
- Set stop freq: \texttt{[:SENSe]:FREQuency:STOP}
- Read X-axis values: \texttt{[:SENSe]:FREQuency:DATA?}
- Set resolution (number of points): \texttt{[:SENSe]:SWEep:POINts}
- Set sweep time: \texttt{[:SENSe]:SWEep:TIME}
<table>
<thead>
<tr>
<th>Command</th>
<th>Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read sweep time</td>
<td>[:SENSe]:SWEep:MTIMe?</td>
</tr>
<tr>
<td>Set manual source power</td>
<td>SOURce:POWer</td>
</tr>
<tr>
<td>Set flat source power</td>
<td>SOURce:POWer:ALC[:MODE]</td>
</tr>
<tr>
<td><strong>Averaging</strong></td>
<td></td>
</tr>
<tr>
<td>Averaging</td>
<td>[:SENSe]:AVERage:COUNT</td>
</tr>
<tr>
<td>Clear Averaging</td>
<td>[:SENSe]:AVERage:CLEAR</td>
</tr>
<tr>
<td>Image rejection</td>
<td>CALCulate:IREJection:LEVEL</td>
</tr>
<tr>
<td>Smoothing On/Off</td>
<td>CALCulate[:SELECTed]:SMOothing[:STATe]</td>
</tr>
<tr>
<td>Smoothing aperture</td>
<td>CALCulate[:SELECTed]:SMOothing:APERture</td>
</tr>
<tr>
<td><strong>Display Items</strong></td>
<td></td>
</tr>
<tr>
<td>View Memory trace</td>
<td>DISPlay:WINDow:TRACe:MEMory:STATe</td>
</tr>
<tr>
<td>View Data trace</td>
<td>DISPlay:WINDow:TRACe:STATe</td>
</tr>
<tr>
<td>Scaling - auto</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO</td>
</tr>
<tr>
<td>Scaling - Set bottom of scale</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:BOTTOM</td>
</tr>
<tr>
<td>Scaling - Set per division</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision</td>
</tr>
<tr>
<td>Scaling - Set reference level</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVEL</td>
</tr>
<tr>
<td>Scaling - Set reference position</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSITION</td>
</tr>
<tr>
<td>Scaling - Set top of scale</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:TOP</td>
</tr>
<tr>
<td><strong>Limit Lines</strong></td>
<td></td>
</tr>
<tr>
<td>Limit line beep</td>
<td>CALCulate[:SELECTed]:LIMit:SOUNd</td>
</tr>
<tr>
<td>Create limit lines</td>
<td>CALCulate[:SELECTed]:LIMit:LLData</td>
</tr>
<tr>
<td>Limit line testing state</td>
<td>CALCulate[:SELECTed]:LIMit:STATe</td>
</tr>
<tr>
<td>Limit line testing annotation</td>
<td>CALCulate[:SELECTed]:LIMit:WARN</td>
</tr>
<tr>
<td>Read Pass / Fail</td>
<td>STATus:QUESTionable:LIMit:CONDition?</td>
</tr>
<tr>
<td>Build Limit from Trace</td>
<td>None</td>
</tr>
<tr>
<td><strong>Markers</strong></td>
<td></td>
</tr>
<tr>
<td>Activate a marker</td>
<td>CALCulate[:SELECTed]:MARKer:ACTivate</td>
</tr>
<tr>
<td>Markers - all off</td>
<td>CALCulate[:SELECTed]:MARKer:AOFF</td>
</tr>
<tr>
<td>Markers - coupled</td>
<td>CALCulate[:SELECTed]:MARKer:COUPled</td>
</tr>
<tr>
<td>Marker search - Max</td>
<td>CALCulate[:SELECTed]:MARKer:FUNCTION:MAXimum</td>
</tr>
<tr>
<td>Marker search - Min</td>
<td>CALCulate[:SELECTed]:MARKer:FUNCTION:MINimum</td>
</tr>
<tr>
<td>Marker search - Peak excursion</td>
<td>CALCulate[:SELECTed]:MARKer:FUNCTION:PEXCursion</td>
</tr>
</tbody>
</table>
Marker search - Peak Next: `CALCulate[:SELection]:MARKer:FUNCtion:PNEXT`
Marker search - Peak threshold: `CALCulate[:SELection]:MARKer:FUNCtion:PTHReshold`
Tracking On/Off: `CALCulate[:SELection]:MARKer:FUNCtion:TRACking`
FieldFox setting => to marker location: `CALCulate[:SELection]:MARKer:SET`
Marker On/Off: `CALCulate[:SELection]:MARKer:[STATE]`
Marker => specified trace: `CALCulate[:SELection]:MARKer:TRACe`
Marker => specified X-axis location: `CALCulate[:SELection]:MARKer:X`
Read Marker Y-axis location: `CALCulate[:SELection]:MARKer:Y?`

Math
Math function: `CALCulate[:SELection]:MATH:FUNCtion`
Data to Memory: `CALCulate[:SELection]:MATH:MEMorize`

Distance to Fault
DTF - Set bandpass/lowpass: `CALCulate:TRANsform:DISTance:BANDpass`
DTF - Set center freq: `CALCulate:TRANsform:DISTance:FREQuency:CENTer`
DTF - Set max freq span: `CALCulate:TRANsform:DISTance:FREQuency:SPAN:MAXimum`
DTF - Set min start freq: `CALCulate:TRANsform:DISTance:FREQuency:STARt:MINimum`
DTF - Set start distance: `CALCulate:TRANsform:DISTance:STARt`
DTF - Set stop distance: `CALCulate:TRANsform:DISTance:STOP`
DTF - Set distance units: `CALCulate:TRANsform:DISTance:UNIT`
DTF - Set window type: `CALCulate:TRANsform:DISTance:WINDow`
DTF - Set bandpass/lowpass: `CALCulate:TRANsform:FREQuency[:TYPE]`

DTF Cable Specs
Select Auto or Manual setting: `[:SENSe]:CORRection:COAX`
Set cable loss: `[:SENSe]:CORRection:LOSS:COAX`
Set velocity factor: `[:SENSe]:CORRection:RVELocity:COAX`

TDR
Set resolution mode: `[:SENSe]:SWEep:TDR:RES`
Max stop frequency with Res mode = AUTO: `[:SENSe]:SWEep:TDR:AUTO:FREQuency:STOP:MAXimum`
TDR marker format: `CALCulate[:SELection]:MARKer:TDR:FORMat`

Save / Recall Files
Recall Cable data  
**MMEMory:LOAD:CABLe**

Store Cable data  
**MMEMory:STORe:CABLE**

Save data trace to csv file  
**MMEMory:STORe:FDATA**

Save SNP data  
**MMEMory:STORe:SNP[:DATA]**

*See other Save / Recall commands*

**Send / Read Data**

Send and read formatted measured data  
**CALCulate[:SELected]:DATA:FDATA**

Send and read formatted memory data  
**CALCulate[:SELected]:DATA:FMEM**

Set data format for read  
**FORMat[:DATA]**

**Calibration Commands**

*See Also*

Correction Methods Explained

Calibration Examples

Set and read error term data  
[:SENSe]:CORRection:COEFFicient[:DATA]

Read number of cal steps  
[:SENSe]:CORRection:COLLect:GUIDed:SCOunt

Measure step number  
[:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire

Prompt for step number  
[:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMp

Measure Quick Cal  
[:SENSe]:CORRection:COLLect[:ACQuire]:INT

Measure load  
[:SENSe]:CORRection:COLLect[:ACQuire]:LOAD

Measure open  
[:SENSe]:CORRection:COLLect[:ACQuire]:OPEN

Measure short  
[:SENSe]:CORRection:COLLect[:ACQuire]:SHORT

Measure thru  
[:SENSe]:CORRection:COLLect[:ACQuire]:THRU

Set Cal Kit  
[:SENSe]:CORRection:COLLect:CKIT:LABel

Catalog all cal kits  
[:SENSe]:CORRection:COLLect:CKIT:LABel:CATalog?

Set connectors  
[:SENSe]:CORRection:COLLect:CONNector

Select method - Enhanced Response  
[:SENSe]:CORRection:COLLect:METHOD:ERES

Select method - QuickCal  
[:SENSe]:CORRection:COLLect:METHOD:QCALibrate:CALibrate

Select method - QuickCal Enhanced Response  
[:SENSe]:CORRection:COLLect:METHOD:QCALibrate:ERESpnone

Select method - Simple Open response  
[:SENSe]:CORRection:COLLect:METHOD[:RESPonse]:OPEN
Select method - Simple Short response
[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:SHORT

Select method - Thru response
[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:THRU

Select method - Short response
[:SENSe]:CORRection:COLLect:METHod:SRESponse

Select method - Open response
[:SENSe]:CORRection:COLLect:METHod:ORESponse

Select method - 1-port SOLT
[:SENSe]:CORRection:COLLect:METHod:SOLT1

Select method - TRL
[:SENSe]:CORRection:COLLect:METHod:TRL

Read method
[:SENSe]:CORRection:COLLect:METHod:TYPE?

Set AutoOrient for ECal
[:SENSe]:CORRection:COLLect:ECAL:AORient

Set simple ECal
[:SENSe]:CORRection:COLLect:ECAL:SIMPle

Omit Isolation
[:SENSe]:CORRection:COLLect:OISolation

Select Medium
[:SENSe]:CORRection:MEDium

Set Waveguide cutoff
[:SENSe]:CORRection:WGCutoff

Select Waveguide standard (CAT only)
[:SENSe]:CORRection:WAVeguide:STANdard

Finish Cal
[:SENSe]:CORRection:COLLect:SAVE

Turn ALL Correction ON and OFF
[:SENSe]:CORRection:[STATe]

Turn User Correction ON and OFF
[:SENSe]:CORRection:USER:[STATe]

Set system impedance
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]

Set CalReady type
[:SENSe]:CORRection:CALReady:TYPE

---

Last Modified:

20-Jan-2015  Added 8.0 commands
15-Nov-2013  Added TRL

Added new commands for A.07.25
Added several commands (A.06.03)
Updated

---

**NA Mode Commands**

63
In this topic:

- **Traces**
- **Sweep Settings**
- **IFBW / Averaging**
- **Display Items**
- **Limit Lines**
- **Markers**
- **Big Marker Readout**
- **Math**
- **Save / Recall Files - Data**
- **Port Extensions**
- **Calibration**
- **Time Domain** (Opt 010)
- **Time Domain Gating** (Opt 010)

**See Also**
- Commands that are Common to All Modes
- Correction Methods Explained
- New Programming Commands
- Status Registers

### Traces

<table>
<thead>
<tr>
<th>Description</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Measurement</td>
<td><code>CALCulate:PARameter{1:4}:DEFine</code></td>
</tr>
<tr>
<td>Set and read number of traces</td>
<td><code>CALCulate:PARameter:COUNt</code></td>
</tr>
<tr>
<td>Select Measurement</td>
<td><code>CALCulate:PARameter{1:4}:SELet</code></td>
</tr>
<tr>
<td>Set trace format</td>
<td><code>CALCulate[:SELECTed]:FORMAT</code></td>
</tr>
<tr>
<td>Multi-trace Configurations</td>
<td><code>DISPlay:WINDow:SPLit</code></td>
</tr>
<tr>
<td>Perform measurement conversion</td>
<td><code>CALCulate[:SELECTed]:CONVersion:FUNCtion</code></td>
</tr>
</tbody>
</table>

### Sweep Settings

<table>
<thead>
<tr>
<th>Description</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set center freq</td>
<td><code>[:SENSe]:FREQuency:CENTer</code></td>
</tr>
<tr>
<td>Set freq span</td>
<td><code>[:SENSe]:FREQuency:SPAN</code></td>
</tr>
<tr>
<td>Set start freq</td>
<td><code>[:SENSe]:FREQuency:START</code></td>
</tr>
<tr>
<td>Set stop freq</td>
<td><code>[:SENSe]:FREQuency:STOP</code></td>
</tr>
<tr>
<td>Read X-axis values</td>
<td><code>[:SENSe]:FREQuency:DATA?</code></td>
</tr>
<tr>
<td>Set resolution (number of points)</td>
<td><code>[:SENSe]:SWEep:POINts</code></td>
</tr>
<tr>
<td>Set sweep time</td>
<td><code>[:SENSe]:SWEep:TIME</code></td>
</tr>
</tbody>
</table>
Read sweep time [:SENSe]:SWEep:MTIme?
Set manual source power SOURce:POWer
Set flat source power SOURce:POWer:ALC[:MODE]
Set trigger Internal or External TRIGger:SOURce
Set polarity of external TRIGger:SLOPe

IFBW / Average / Smooth / Image Rej
IFBW [:SENSe]:BWID
Averaging [:SENSe]:AVERage:COUNT
Clear Averaging [:SENSe]:AVERage:CLEar
Average (Swp/Point) [:SENSe]:AVERage:MODE
Smoothing ON/OFF CALCulate[:SELected]:SMOothing[:STATe]
Smoothing aperture CALCulate[:SELected]:SMOothing:APERture

Display Items
View Memory trace DISPlay:WINDow:TRACe:MEMory:STATe
View Data trace DISPlay:WINDow:TRACe:STATe
Scaling - auto DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO
Scaling - Set bottom of scale DISPlay:WINDow:TRACe:Y[:SCALe]:BOTTom
Scaling - Set per division DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision
Scaling - Set reference level DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVEL
Scaling - Set reference position DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSition
Scaling - Set top of scale DISPlay:WINDow:TRACe:Y[:SCALe]:TOP
Electrical Delay CALCulate[:SELected]:CORRection:EDELay:TIME
Phase Offset CALCulate[:SELected]:CORRection:OFFSet:PHASe
Mag Offset CALCulate[:SELected]:OFFSet:MAGNitude
Mag Slope CALCulate[:SELected]:OFFSet:SLOPe

Limit Lines
Limit line beep CALCulate[:SELected]:LIMit:SOUNd
Create limit lines CALCulate[:SELected]:LIMit:LLData
Limit line testing state CALCulate[:SELected]:LIMit:STATe
Limit line testing annotation CALCulate[:SELected]:LIMit:WARN
Read Pass / Fail STATus:QUEtionable:LIIMit:CONDition?
Build Limit from Trace

**Markers**

Marker On/Off  \( \text{CALCulate[:SELECTed]:MARKer[:STATE]} \)

Activate a marker  \( \text{CALCulate[:SELECTed]:MARKer:ACTivate} \)

Markers - all off  \( \text{CALCulate[:SELECTed]:MARKer:AOFF} \)

Read BW data  \( \text{CALCulate[:SELECTed]:MARKer:BWIDth:DATA} \)

Marker search - BW / Q On/Off  \( \text{CALCulate[:SELECTed]:MARKer:BWIDth[:STATE]} \)

Markers - coupled  \( \text{CALCulate[:SELECTed]:MARKer:COUPled} \)

BW search value  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:BWIDth:THReshold} \)

Marker search - Max  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:MAXimum} \)

Marker search - Min  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:MINimum} \)

Marker search - Peak excursion  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:PEXCursion} \)

Marker search - Peak Next  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:PNEXT} \)

Marker search - Peak threshold  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:PTHReshold} \)

Marker search - Target  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:TARGet} \)

Marker search - Wrap/No Wrap  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:TDIRection} \)

Tracking On/Off  \( \text{CALCulate[:SELECTed]:MARKer:FUNCtion:TRACking} \)

FieldFox setting => to marker location  \( \text{CALCulate[:SELECTed]:MARKer:SET} \)

Marker format  \( \text{CALCulate[:SELECTed]:MARKer:FORMat} \)

Marker => specified trace  \( \text{CALCulate[:SELECTed]:MARKer:TRACe} \)

Marker => specified X-axis location  \( \text{CALCulate[:SELECTed]:MARKer:X} \)

Read Marker Y-axis location  \( \text{CALCulate[:SELECTed]:MARKer:Y?} \)

**Big Marker Display States (A and B)**

Enable Big marker readouts  \( \text{DISPlay:MARKer:LARGE:STATE} \)

Set font size  \( \text{DISPlay:MARKer:LARGE:<x>:FONT} \)

Set how the marker is displayed (norm / delta)  \( \text{DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:MARKer:STATE} \)

Enable a readout line  \( \text{DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:STATE} \)

Assign a marker number to the readout line  \( \text{DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:MNUMber} \)

Assign a trace number to the  \( \text{DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:TNUMber} \)
readout line
Set format for the readout line DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:FORMat
Set marker tracking DISPlay:MARKer:LARGE:<x>:DEFine:LINE<n>:TRACking
Set number of traces DISPlay:MARKer:LARGE:<x>:TRACE:COUNt
Set measurement for the trace DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:MEASurement
Set format for the trace DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:FORMat
Enable bandwidth search DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:8WIDth:STATe

Math
Math function CALCulate[:SElected]:MATH:FUNCtion
Data to Memory CALCulate[:SElected]:MATH:MEMorize

Save / Recall Files - Data
Save data trace to csv file MMEMory:STORe:FDATa
Save SNP data MMEMory:STORe:SNP[:DATA]

See other Save / Recall commands
Send and read formatted measured data CALCulate[:SElected]:DATA:FDATa
Send and read formatted memory data CALCulate[:SElected]:DATA:FMEM
Send and read unformatted measured data CALCulate[:SElected]:DATA:SDATa
Send and read unformatted memory data CALCulate[:SElected]:DATA:SMEM
Set read format FORMat[:DATA]

Port Extensions
Enable [:SENSe]:CORRection:EXTension[:STATe]
Set port 1 [:SENSe]:CORRection:EXTension:PORT1
Set port 2 [:SENSe]:CORRection:EXTension:PORT2
Set velocity factor [:SENSe]:CORRection:RVELocity:COAX

Calibration
See Also
Correction Methods Explained
Calibration Examples
Set and read error term data [:SENSe]:CORRection:COEFFicient[:DATA]
Read number of cal steps [:SENSe]:CORRection:COLLection:GUIDed:SCOunt
Measure step number [:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire
Prompt for step number [:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMpt
Measure Quick Cal [:SENSe]:CORRection:COLLect:ACQuire:INT
Measure load [:SENSe]:CORRection:COLLect:ACQuire:LOAD
Measure open [:SENSe]:CORRection:COLLect:ACQuire:OPEN
Measure short [:SENSe]:CORRection:COLLect:ACQuire:SHORT
Measure thru [:SENSe]:CORRection:COLLect:ACQuire:THRU
Set Cal Kit [:SENSe]:CORRection:COLLect:CKIT:LABel
Catalog all cal kits [:SENSe]:CORRection:COLLect:CKIT:LABel:CATalog?
Set connectors [:SENSe]:CORRection:COLLect:CONNector
Select method - Enhanced [:SENSe]:CORRection:COLLect:METHod:ERES
Select method - QuickCal [:SENSe]:CORRection:COLLect:METHod:QCALibrate:CALibrate
Select method - QuickCal Enhanced Response [:SENSe]:CORRection:COLLect:METHod:QCALibrate:ERESponse
Select method - Simple Open response [:SENSe]:CORRection:COLLect:METHod:[RESPonse]:OPEN
Select method - Simple Short response [:SENSe]:CORRection:COLLect:METHod:[RESPonse]:SHORT
Select method - Thru response [:SENSe]:CORRection:COLLect:METHod:[RESPonse]:THRU
Select method - Open response [:SENSe]:CORRection:COLLect:METHod:[RESPonse]:ORESponse
Select method - Short response [:SENSe]:CORRection:COLLect:METHod:[RESPonse]:SRESponse
Select method - 1-port SOLT [:SENSe]:CORRection:COLLect:METHod:SOLT1
Select method - TRL [:SENSe]:CORRection:COLLect:METHod:TRL
Read method [:SENSe]:CORRection:COLLect:METHod:TYPE?
Set AutoOrient for ECal [:SENSe]:CORRection:COLLect:ECAL:AORient
Set simple ECal [:SENSe]:CORRection:COLLect:ECAL:SIMPle
Omit Isolation [:SENSe]:CORRection:COLLect:OISolation
Select Medium [:SENSe]:CORRection:MEDium
Set Waveguide cutoff [:SENSe]:CORRection:WGcutoff
Finish Cal [:SENSe]:CORRection:COLLect:SAVE
Turn ALL Correction ON and OFF
[:SENSe]:CORRection[:STATe]

Turn User Correction ON and OFF
[:SENSe]:CORRection:USER[:STATe]

Set system impedance
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]

Set CalReady type
[:SENSe]:CORRection:CALReady:TYPE

**Time Domain (Opt 010)**

Enable
CALCulate[:SELected]:TRANsform:TIME:STATE

Start time
CALCulate[:SELected]:TRANsform:TIME:START

Stop time
CALCulate[:SELected]:TRANsform:TIME:STOP

Center time
CALCulate[:SELected]:TRANsform:TIME:CENTer

Span time
CALCulate[:SELected]:TRANsform:TIME:SPAN

Lowpass or BandPass
CALCulate[:SELected]:TRANsform:TIME: [TYPE]

Step or Impulse
CALCulate[:SELected]:TRANsform:TIME:STIMulus

Step rise time
CALCulate[:SELected]:TRANsform:TIME:STEP:RTIMe

Impulse width
CALCulate[:SELected]:TRANsform:TIME:IMPulse:WIDTh

Kaiser Bessel width
CALCulate[:SELected]:TRANsform:TIME:KBESsel

Set Lowpass freq.
CALCulate[:SELected]:TRANsform:TIME:LPFREQuency

**Time Domain Gating**

Enable
CALCulate[:SELected]:FILTer[:GATE]:TIME:STATE

Start time
CALCulate[:SELected]:FILTer[:GATE]:TIME:START

Stop time
CALCulate[:SELected]:FILTer[:GATE]:TIME:STOP

Center time
CALCulate[:SELected]:FILTer[:GATE]:TIME:CENTer

Span time
CALCulate[:SELected]:FILTer[:GATE]:TIME:SPAN

Gating shape
CALCulate[:SELected]:FILTer[:GATE]:TIME:SHApe

Bandpass or notch
CALCulate[:SELected]:FILTer[:GATE]:TIME

---

**Last Modified:**

13-Nov-2013   Added TRL command

6-Mar-2013     Added new commands (A.06.25)

18-Oct-2012    Added new commands (A.06.00)
SA Mode Commands

In this topic:

- Frequency
- Radio Standard
- Gain / Atten
- Sweep
- Scale/Units
- Average
- Video / Res Bandwidth
- Trace Type / Detector
- Alignments (InstAlign)
- Limit / Display Lines
- Markers
- Read / Save Data
- Independent Source / Tracking Generator
- Source Tracking Offset
- Field Strength (Corrections)
- Trigger Settings
- FFT Gating
- Record/Playback Actions
- Record/Playback Configuration
- Record/Playback Sessions
- Select Channel Measurement
- Read Current Channel Measurement Data
- Channel Measurement Setup
- Adjacent Channel Power Setup
- Interference Analyzer Settings
- Tune and Listen Settings
- General Status

See Also
- Commands that are Common to All Modes
- Status Registers

### Frequency

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center freq</td>
<td>[:SENSe]:FREQuency:CENTer</td>
</tr>
<tr>
<td>Step size for up/down keys</td>
<td>[:SENSe]:FREQuency:CENTer:STEP</td>
</tr>
<tr>
<td>Step size auto/manual</td>
<td>[:SENSe]:FREQuency:CENTer:STEP:AUTO</td>
</tr>
<tr>
<td>Feature</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Freq span</td>
<td>[:SENSe]:FREQuency:SPAN</td>
</tr>
<tr>
<td>Span to full</td>
<td>[:SENSe]:FREQuency:SPAN:FULL</td>
</tr>
<tr>
<td>Span to zero</td>
<td>[:SENSe]:FREQuency:SPAN:ZERO</td>
</tr>
<tr>
<td>Start freq</td>
<td>[:SENSe]:FREQuency:STARt</td>
</tr>
<tr>
<td>Stop freq</td>
<td>[:SENSe]:FREQuency:STOP</td>
</tr>
<tr>
<td>Frequency annotation method</td>
<td>[:SENSe]:FREQuency:ANNotation[:SElECT]</td>
</tr>
<tr>
<td><strong>Radio Standard</strong></td>
<td></td>
</tr>
<tr>
<td>Channel number center</td>
<td>[:SENSe]:RADio:CHANnel:CENTer</td>
</tr>
<tr>
<td>Uplink or Downlink</td>
<td>[:SENSe]:RADio:CHANnel:DIRection</td>
</tr>
<tr>
<td>Start channel</td>
<td>[:SENSe]:RADio:CHANnel:STARt</td>
</tr>
<tr>
<td>Channel step size</td>
<td>[:SENSe]:RADio:CHANnel:STEP</td>
</tr>
<tr>
<td>Stop channel</td>
<td>[:SENSe]:RADio:CHANnel:STOP</td>
</tr>
<tr>
<td>Select standard</td>
<td>[:SENSe]:RADio:STANdard[:SElECT]</td>
</tr>
<tr>
<td>Freq or Chan</td>
<td>[:SENSe]:RADio:TEUNit</td>
</tr>
<tr>
<td><strong>Gain/Atten</strong></td>
<td></td>
</tr>
<tr>
<td>Attenuation value</td>
<td>[:SENSe]:POWer[:RF]:ATTenuation</td>
</tr>
<tr>
<td>Atten Auto/Manual</td>
<td>[:SENSe]:POWer[:RF]:ATTenuation:AUTO</td>
</tr>
<tr>
<td>Set external gain value</td>
<td>[:SENSe]:POWer[:RF]:EXTGain</td>
</tr>
<tr>
<td>Preamp ON/OFF</td>
<td>[:SENSe]:POWer[:RF]:GAIN[:STATe]</td>
</tr>
<tr>
<td>Read number of traces.</td>
<td>[:SENSe]:QUANtity:TACTive?</td>
</tr>
<tr>
<td><strong>Sweep</strong></td>
<td></td>
</tr>
<tr>
<td>Sweep time (Non-zerospan)</td>
<td>[:SENSe]:SWEep:ACQuisition</td>
</tr>
<tr>
<td>Sweep time (Zerospan)</td>
<td>[:SENSe]:SWEep:TIME</td>
</tr>
<tr>
<td>Read sweep time</td>
<td>[:SENSe]:SWEep:MTIMe?</td>
</tr>
<tr>
<td>Auto / Manual</td>
<td>[:SENSe]:SWEep:ACQuisition:AUTO</td>
</tr>
<tr>
<td>Sweep points</td>
<td>[:SENSe]:SWEep:POINts</td>
</tr>
<tr>
<td>Sweep Type</td>
<td>[:SENSe]:SWEep:TYPe</td>
</tr>
<tr>
<td>Reverse sweep (Step sweep and Opt. 209 only)</td>
<td>[:SENSe]:FREQuency:SPAN:DREVerse</td>
</tr>
<tr>
<td><strong>Scale / Units</strong></td>
<td></td>
</tr>
<tr>
<td>Auto Scale trace</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO</td>
</tr>
<tr>
<td>Per division</td>
<td>DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision</td>
</tr>
<tr>
<td>Command</td>
<td>Command String</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Reference level</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALE]:RLEVel</td>
</tr>
<tr>
<td>Reference position</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALE]:RPOsition</td>
</tr>
<tr>
<td>Log / Linear scale</td>
<td>[:SENSe]:AMPLitude:SCALE</td>
</tr>
<tr>
<td>Set Units</td>
<td>[:SENSe]:AMPLitude:UNIT</td>
</tr>
<tr>
<td>Read ONLY - Bottom scale value</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALE]:BOTTom</td>
</tr>
<tr>
<td>Read ONLY - Top scale value</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALE]:TOP</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
</tr>
<tr>
<td>Average count</td>
<td>[:SENSe]:AVERage:COUNt</td>
</tr>
<tr>
<td>Averaging type</td>
<td>[:SENSe]:AVERage:TYPE</td>
</tr>
<tr>
<td>Restart trace averaging</td>
<td>INITiate:RESTart</td>
</tr>
<tr>
<td><strong>Video / Res Bandwidth</strong></td>
<td></td>
</tr>
<tr>
<td>Manual Res BW value</td>
<td>[:SENSe]:BANDwidth[:RESolution]</td>
</tr>
<tr>
<td>Choose Auto or Manual Res BW</td>
<td>[:SENSe]:BANDwidth[:RESolution]:AUTO</td>
</tr>
<tr>
<td>Manual Video BW value</td>
<td>[:SENSe]:BANDwidth:VIDeo</td>
</tr>
<tr>
<td>Choose Auto or Manual Video BW</td>
<td>[:SENSe]:BANDwidth:VIDeo:AUTO</td>
</tr>
<tr>
<td>IF Output</td>
<td>[:SENSe]:BANDwidth:IF:OUT</td>
</tr>
<tr>
<td><strong>Trace Type/Detector</strong></td>
<td></td>
</tr>
<tr>
<td>Set Trace Type</td>
<td>TRACe{1:4}:TYPE</td>
</tr>
<tr>
<td>Detector function</td>
<td>[:SENSe]:DETector:FUNCtion</td>
</tr>
<tr>
<td><strong>Alignments (InstAlign)</strong></td>
<td></td>
</tr>
<tr>
<td>Align all now (coupled to all individual alignments)</td>
<td>[:SENSe]:ALIGNment:ALL:NOW</td>
</tr>
<tr>
<td>Align all state (coupled to all individual alignments)</td>
<td>[:SENSe]:ALIGNment:ALL[:STATe]</td>
</tr>
<tr>
<td>Align now</td>
<td>[:SENSe]:ALIGNment:AMPLitude:NOW</td>
</tr>
<tr>
<td>InstAlign state</td>
<td>[:SENSe]:ALIGNment:AMPLitude[:STATe]</td>
</tr>
<tr>
<td>RF burst now (Not applicable to ERTA)</td>
<td>[:SENSe]:ALIGNment:BURSt:NOW</td>
</tr>
<tr>
<td>RF burst state (Not applicable to ERTA)</td>
<td>[:SENSe]:ALIGNment:BURSt[:STATe]</td>
</tr>
<tr>
<td><strong>Limit / Display Lines</strong></td>
<td></td>
</tr>
<tr>
<td>Limit line beep</td>
<td>CALCulate[:SELe]LIMit:SOUNd</td>
</tr>
<tr>
<td>Create limit lines</td>
<td>CALCulate:LIMit:LLData</td>
</tr>
<tr>
<td>Limit line testing state</td>
<td>CALCulate:LIMit[:STATe]</td>
</tr>
<tr>
<td>Feature</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Limit line testing annotation</td>
<td>CALCulate:LIMit:WARN</td>
</tr>
<tr>
<td>Display Line level setting</td>
<td>DISPLAY:WINDow:TRACe:Y:DLINe</td>
</tr>
<tr>
<td>Display Line state</td>
<td>DISPLAY:WINDow:TRACe:Y:DLINe:STATe</td>
</tr>
<tr>
<td>Read Pass/Fail</td>
<td>STATus:QUESTIONable:LIMit:CONDition?</td>
</tr>
<tr>
<td>Build Limit from Trace</td>
<td>None</td>
</tr>
</tbody>
</table>

**Markers**

- Select a marker: CALCulate:MARKer:ACTivate
- Markers - all off: CALCulate:MARKer:AOFF
- Audio Beep on Marker: CALCulate:MARKer:AUDio:BEEP
- Frequency counter marker ON/OFF: CALCulate:MARKer:FCOunt[STATe]
- Read Frequency Count: CALCulate:MARKer:FCOunt:X?
- Set Noise Marker and Band Power Marker: CALCulate:MARKer:FUNCTION

- Band power span: CALCulate:MARKer:FUNCTION:BAND:SPAN
- Band power mode: CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO
- Interval power span: CALCulate:MARKer:FUNCTION:INTerval:SPAN
- Interval power mode: CALCulate:MARKer:FUNCTION:INTerval:SPAN:AUTO
- Marker search - Max: CALCulate:MARKer{1:6}:FUNCTION:MAXimum
- Marker search - Min: CALCulate:MARKer{1:6}:FUNCTION:MINimum
- Marker search - Peak excursion: CALCulate:MARKer{1:6}:FUNCTION:PEXCursion
- Marker search - Peak left: CALCulate:MARKer{1:6}:FUNCTION:PLEFt
- Marker search - Peak Next: CALCulate:MARKer{1:6}:FUNCTION:PNEXt
- Marker search - Peak right: CALCulate:MARKer{1:6}:FUNCTION:PRIGht
- Marker search - Peak threshold: CALCulate:MARKer{1:6}:FUNCTION:PTHReshold
- Noise marker On/Off: CALCulate:MARKer:NOISe[STATe]
- Marker -> Setting: CALCulate:MARKer:SET
- Move marker to center freq: CALCulate:MARKer:SET:CENTer
- Move marker to ref level: CALCulate:MARKer:SET:REFLevel
- Signal Tracking: CALCulate:MARKer:STRack
Commands by Mode

Marker On/Off

CALCulate:MARKer[:STATe]

Move marker to other trace

CALCulate:MARKer:TRACe

Move/read marker X-axis position

CALCulate:MARKer:X

Read marker Y-axis position

CALCulate:MARKer:Y?

Read / Save Data

See Also: Read Current Channel Measurement Data

Read Trace Data

TRACe{1:4}:DATA

Saves trace to CSV file.

MMEMory:STORe:FDATa

Independent Source / Tracking Generator

CW or S/R

SOURce:MODE

CW Frequency

SOURce:FREQuency[:CW]

Normalize

SOURce:NORMalize

Power Level (All models EXCEPT N9912A)

SOURce:POWer

Max Power (All models EXCEPT N9912A)

SOURce:POWer:MAXimum

Attenuation (power) N9912A ONLY

SOURce:POWer:ATTenuation

Enable Source Tracking

SOURce:TRACking

Source Tracking Offset

Enable

SOURce:TOFFset:ENABLE

Set frequency

SOURce:TOFFset:FREQuency

Reverse frequency

SOURce:TOFFset:REVerse

Field Strength (Corrections)

All corrections ON

[:SENSe]:AMPLitude:CORRections[:STATe]

Receiver (or ONLY one) Antenna/Cables

Antenna corrections ON

[:SENSe]:AMPLitude:CORRections:ANTenna[:STATe]

Cable corrections ON

[:SENSe]:AMPLitude:CORRections:CABLE[:STATe]

Load Antenna file

MMEMory:LOAD:ANTenna

Store Antenna file

MMEMory:STORe:ANTenna

Clear Antenna correction values

[:SENSe]:AMPLitude:CORRections:ANTenna:DEFault

Load Cable file

MMEMory:LOAD:CABLE

Store Cable file

MMEMory:STORe:CABLE
Clear Cable correction values [:SENSe]:AMPLitude:CORRections:CABLe:DEFault

**Source Antenna/Cables**

Antenna corrections ON [:SENSe]:AMPLitude:CORRections:SANTenna:[STATe]
Cable corrections ON [:SENSe]:AMPLitude:CORRections:SCABle:[STATe]
Load Antenna file MMEMory:LOAD:SANTenna
Store Antenna file MMEMory:STORE:SANTenna
Clear Antenna correction values [:SENSe]:AMPLitude:CORRections:SANTenna:DEFault
Load Cable file MMEMory:LOAD:SCABle
Store Cable file MMEMory:STORE:SCABle
Clear Cable correction values [:SENSe]:AMPLitude:CORRections:SCABle:DEFault

**Trigger Settings**

Trigger Type TRIGger[:SEQuence]:SOURce
Trigger Slope (Pos/Neg) TRIGger[:SEQuence]:SLOPe
Trigger Delay TRIGger[:SEQuence]:DELay
Trigger Delay ON/OFF TRIGger[:SEQuence]:DELay:STATe
Trigger Level TRIGger[:SEQuence]:VIDeo:LEVel
Auto trigger time TRIGger[:SEQuence]:ATRigger
Auto trigger ON/OFF TRIGger[:SEQuence]:ATRigger:STATe
Trigger Position TRIGger[:SEQuence]:POSition
Trigger Position ON/OFF TRIGger[:SEQuence]:POSition:STATe

**FFT Gating**

Enable gating TRIGger[:SEQuence]:FGATe:[STATe]
Set gate delay TRIGger[:SEQuence]:FGATe:DELay
Display the gating window TRIGger[:SEQuence]:FGATe:VIEW:[STATe]
Set X-axis time span for the gating window TRIGger[:SEQuence]:FGATe:VIEW:TIME
Set the width of the gated area within the gating window TRIGger[:SEQuence]:FGATe:WIDTh

**Record/Playback Actions**

Pause RECPlayback:ACTion:PAUSe
Play RECPlayback:ACTion:PLAY
Record RECPlayback:ACTion:RECorD
<table>
<thead>
<tr>
<th>Tag position</th>
<th>RECPlayback:ACTion:SPOsition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>RECPlayback:ACTion:STOP</td>
</tr>
<tr>
<td>Trace position</td>
<td>RECPlayback:ACTion:TPOSition</td>
</tr>
</tbody>
</table>

**Record/Playback Configuration**

<table>
<thead>
<tr>
<th>Frequency Mask Trigger (FMT) data</th>
<th>RECPlayback:CONFig:FMTRigger:LLData</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMT enable</td>
<td>RECPlayback:CONFig:FMTRigger:ENABLE</td>
</tr>
<tr>
<td>Playback Time Interval</td>
<td>RECPlayback:CONFig:PTInterval</td>
</tr>
<tr>
<td>Record Segment Counting Length</td>
<td>RECPlayback:CONFig:RSCLength</td>
</tr>
<tr>
<td>Record Source</td>
<td>RECPlayback:CONFig:RSOurce</td>
</tr>
<tr>
<td>Record Time Interval</td>
<td>RECPlayback:CONFig:RTINerval</td>
</tr>
<tr>
<td>Record Time limit</td>
<td>RECPlayback:CONFig:RTLSeconds</td>
</tr>
</tbody>
</table>

**Record/Playback Sessions**

| Clear all records                | RECPlayback:SESSion:CARecords     |
| Close                             | RECPlayback:SESSion:CLOSE         |
| Create new session               | RECPlayback:SESSion:NEW           |
| Open session                     | RECPlayback:SESSion:OPEN?         |
| Storage device                    | RECPlayback:SESSion:SDEVICE       |
| Trace record limit               | RECPlayback:SESSion:TRLimit       |
| Is a session open?               | STATUS:OPERation:SMode:CONDITION? |

**Select Channel Measurement**

| All channel meas OFF             | [:SENSe]:MEASurement:AOFF         |
| Select channel meas              | [:SENSe]:MEASurement:CHANnel      |
| Select Interference Analysis (Opt 236) display. | [:SENSe]:MEASurement:INTerference |
| Preset channel meas              | [:SENSe]:MEASurement:PRESet       |
| Tune and Listen demod type       | [:SENSe]:MEASurement:TLListen     |
| Power percent for Occupied Bandwidth | [:SENSe]:OBW:POW                  |

**Read Current Channel Measurement Data**

Channel Power, Occupied Bandwidth, or Adjacent Channel Power

| Read current channel measurement data | CALCulate:MEASurement:DATA? |

**Channel Measurement Setup**


Averaging ON/OFF | [:SENSe]:CMEasurement:AVERage:ENABLE
Integration BW | [:SENSe]:CMEasurement:IBW
RCC filter state | [:SENSe]:CMEasurement:RRCFilter
RCC filter value | [:SENSe]:CMEasurement:RRCFilter:ALPHA

**Adjacent Channel Power Setup**
Reference value | [:SENSe]:ACPower:MREFERENCE
Reference method | [:SENSe]:ACPower:MREFERENCE:AUTO
Meas Type | [:SENSe]:ACPower:MTYPE
Limit testing On/Off | [:SENSe]:ACPower:LIMIT[:STATE]
Offset bandwidths | [:SENSe]:ACPower:OFFSet:BWIDth
Offset frequencies | [:SENSe]:ACPower:OFFSet:FREQuency
Offset states | [:SENSe]:ACPower:OFFSet:STATe
Lower offset limits | [:SENSe]:ACPower:OFFSet:LLIMit
Upper offset limits | [:SENSe]:ACPower:OFFSet:ULIMit

**Interference Analyzer Settings**
Blue power level | [:SENSe]:SPECtrogram:BPLevel
Red power level | [:SENSe]:SPECtrogram:RPLevel
Time/Delta marker state | [:SENSe]:SPECtrogram:TMARker:STATe
Time/Delta marker location | [:SENSe]:SPECtrogram:TMARker:VALue
Spectrogram display setting | [:SENSe]:SPECtrogram:VIEW
Waterfall angle | [:SENSe]:SPECtrogram:WANGle

**Tune and Listen Settings**
Volume | [:SENSe]:TALListen:AVOLume- OBSOLETE
Demod ON / OFF | [:SENSe]:TALListen:DSTate
Demod type | [:SENSe]:TALListen:DTYPe
Listen tme | [:SENSe]:TALListen:LTIMe
Tune freq | [:SENSe]:TALListen:TFReq

**General Status**
ADC Over-range status | CALCulate:MEASurement:WAOR?

---

Last Modified:

19-sep-2016  Corrected/Added SA reverse sweep SCPI command (step sweep only). (:FREQ:SPAN:DREV)
RTSA Mode Commands (A.90.xx and Greater Firmware Only)

In this topic:
- Frequency
- Gain/Atten
- Sweep
- Scale/Units
- Average
- Res Bandwidth
- Trace Type/Detector
- Alignments (InstAlign)
- Limit / Display Lines
- Markers
- Read / Save Data
- Source
- Trigger Settings
- Record/Playback Actions
- Record/Playback Configuration
- Record/Playback Sessions
- Density
- Spectogram
- See Also

### Commands that are Common to All Modes

**Frequency**

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center freq</td>
<td>[:SENSe]:FREQuency:CENTer</td>
</tr>
<tr>
<td>Step size for up/down keys</td>
<td>[:SENSe]:FREQuency:CENTer:STEP</td>
</tr>
<tr>
<td>Step size auto/manual</td>
<td>[:SENSe]:FREQuency:CENTer:STEP:AUTO</td>
</tr>
<tr>
<td>Freq span</td>
<td>[:SENSe]:FREQuency:SPAN</td>
</tr>
<tr>
<td>Span to full</td>
<td>[:SENSe]:FREQuency:SPAN:FULL</td>
</tr>
<tr>
<td>Start freq</td>
<td>[:SENSe]:FREQuency:STARt</td>
</tr>
<tr>
<td>Stop freq</td>
<td>[:SENSe]:FREQuency:STOP</td>
</tr>
</tbody>
</table>
Frequency annotation method

Gain/Attenuation
- Attenuation value
- Attenuation value [SENSe]:POWer[:RF]:ATTenuation
- Attenuation Auto/Manual
- Atten Auto/Manual [:SENSe]:POWer[:RF]:ATTenuation:AUTO
- Set external gain value
- Set external gain value [:SENSe]:POWer[:RF]:EXTGain
- Preamp ON/OFF/AUTO
- Preamp ON/OFF/AUTO [:SENSe]:POWer[:RF]:GAIN[:STATe]

Sweep
- Selects between Density, Spectrogram, & Real-Time Spectrum traces
- Selects the acquisition time in the Density & Real-time Traces
- Sets the acquisition time in the Density & Real-time Traces to AUTO

Scale / Units
- Per division
  - Per division DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:PDIVision
- Reference level
  - Reference level DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:RLEVEL
- Reference position
  - Reference position DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:RPOSITION

Average
-平均计数
- Average count [:SENSe]:AVERage:COUNt
- Restart trace averaging
- Restart trace averaging INITiate:RESTart

Res Bandwidth
- Manual Res BW value
  - Manual Res BW value [:SENSe]:BANDwidth[:RESolution]
- Choose Auto or Manual Res BW
  - Choose Auto or Manual Res BW [:SENSe]:BANDwidth[:RESolution]:AUTO
- Query ratio of Span to Res BW
  - Query ratio of Span to Res BW [:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?

Trace Type/Detector
- Set Trace Type
  - TRACe{1:4}:TYPE
- Detector function
  - [:SENSe]:DETector:TRACe{1:4}:FUNCtion
- Retrieves trace data
  - TRACe:DATA?
- Defaults all traces back to their default state
  - TRACe:PRESet:ALL

Alignments (InstAlign)
- Align all now (coupled to all individual alignments)
  - Align all now [:SENSe]:ALIgnment:ALL:NOW
- Align all state (coupled to all individual alignments)
  - Align all state [:SENSe]:ALIgnment:ALL[:STATe]
### Commands by Mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align now</td>
<td>[:SENSe]:ALIGnment:AMPLitude:NOW</td>
</tr>
<tr>
<td>InstAlign state</td>
<td>[:SENSe]:ALIGnment:AMPLitude[:STATe]</td>
</tr>
<tr>
<td>RF burst now</td>
<td>[:SENSe]:ALIGnment:BURSt:NOW</td>
</tr>
<tr>
<td>RF burst state</td>
<td>[:SENSe]:ALIGnment:BURSt[:STATe]</td>
</tr>
<tr>
<td>Channel equalization now</td>
<td>[:SENSe]:ALIGnment:CHEQ:NOW</td>
</tr>
<tr>
<td>Channel equalization state</td>
<td>[:SENSe]:ALIGnment:CHEQ[:STATe]</td>
</tr>
</tbody>
</table>

#### Limit / Display Lines

<table>
<thead>
<tr>
<th>Command</th>
<th>Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Line level setting</td>
<td>DISPLAY:WINDow:TRACe:Y:DLINe</td>
</tr>
<tr>
<td>Display Line state</td>
<td>DISPLAY:WINDow:TRACe:Y:DLINe[:STATe]</td>
</tr>
</tbody>
</table>

#### Markers

<table>
<thead>
<tr>
<th>Command</th>
<th>Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a marker</td>
<td>CALCulate:MARKer{1:6}:ACTivate</td>
</tr>
<tr>
<td>Markers - all off</td>
<td>CALCulate:MARKer{1:6}:AOFF</td>
</tr>
<tr>
<td>Markers - Fixed delta reference marker state</td>
<td>CALCulate:MARKer{1:6}:DREF:FIXed</td>
</tr>
<tr>
<td>Marker search - Max</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:MAXimum</td>
</tr>
<tr>
<td>Marker search - Min</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:MINimum</td>
</tr>
<tr>
<td>Marker search - Peak excursion</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:PEXCursion</td>
</tr>
<tr>
<td>Marker search - Peak left</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:PLEFT</td>
</tr>
<tr>
<td>Marker search - Peak Next</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:PNEXT</td>
</tr>
<tr>
<td>Marker search - Peak right</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:PRIGHT</td>
</tr>
<tr>
<td>Marker search - Peak threshold</td>
<td>CALCulate:MARKer{1:6}:FUNCtion:PTHReshold</td>
</tr>
<tr>
<td>Move marker to center freq</td>
<td>CALCulate:MARKer{1:6}:SET:CENTer</td>
</tr>
<tr>
<td>Move marker to ref level</td>
<td>CALCulate:MARKer{1:6}:SET:REFLevel</td>
</tr>
<tr>
<td>Marker On/Off</td>
<td>CALCulate:MARKer{1:6}[:STATe]</td>
</tr>
<tr>
<td>Move marker to other trace</td>
<td>CALCulate:MARKer{1:6}:TRACe</td>
</tr>
<tr>
<td>Move/read marker X-axis position</td>
<td>CALCulate:MARKer{1:6}:X</td>
</tr>
<tr>
<td>Read marker Y-axis position</td>
<td>CALCulate:MARKer{1:6}:Y?</td>
</tr>
<tr>
<td>Enable/Disables a displayed marker table</td>
<td>:DISPLAY:TABLE:MARKer</td>
</tr>
</tbody>
</table>

#### Read / Save Data

See Also: Read Current Channel Measurement Data

<table>
<thead>
<tr>
<th>Command</th>
<th>Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Trace Data</td>
<td>TRACe{1:4}:DATA</td>
</tr>
</tbody>
</table>
Saves trace to CSV file. **MMEMory:STORe:FDATA**

**Source**

**Trigger Settings**

- Auto trigger time **TRIGger[:SEQUence]:ATRigger**
- Auto trigger ON/OFF **TRIGger[:SEQUence]:ATRigger:STATe**
- Trigger Delay **TRIGger[:SEQUence]:DELay**
- Trigger Delay ON/OFF **TRIGger[:SEQUence]:DELay:STATe**
- Trigger Frame Offset value **:TRIGger[:SEQUence]:FRAMe:OFFSet**
- Trigger Frame Offset Reset **:TRIGger[:SEQUence]:FRAMe:OFFSet:DISPlay:RESet**
- Set the period of the internal periodic timer clock **:TRIGger[:SEQUence]:FRAMe:PERiod**
- Sets the holdoff time between triggers **:TRIGger[:SEQUence]:HOLDoff**
- Sets the level for the trigger **:TRIGger[:SEQUence]:LEVel**
- Trigger Type (Ext/Vid/RFBurst/Freerun) **TRIGger[:SEQUence]:SOURce**
- Trigger Slope (Pos/Neg) **TRIGger[:SEQUence]:SLOPe**

**Record/Playback Actions**

- Pause **RECPlayback:ACTion:PAUSe**
- Play **RECPlayback:ACTion:PLAY**
- Record **RECPlayback:ACTion:RECord**
- Tag position **RECPlayback:ACTion:SPOSition**
- Stop **RECPlayback:ACTion:STOP**
- Trace position **RECPlayback:ACTion:TPOSition**

**Record/Playback Configuration**

- Playback Time Interval **RECPlayback:CONFig:PTINterval**
- Record Segment Counting Length **RECPlayback:CONFig:RSCLength**
- Record Source **RECPlayback:CONFig:RSOurce**
- Record Time Interval **RECPlayback:CONFig:RTINerval**
- Record Time limit **RECPlayback:CONFig:RTLSeconds**

**Record/Playback Sessions**

- Clear all records **RECPlayback:SESSion:CARecords**
- Close **RECPlayback:SESSion:CLOSE**
- Create new session **RECPlayback:SESSion:NEW**
Commands by Mode

Open session  
Storage device  
Trace record limit  
Is a session open?  

Density
- Density blue limit percentage  
- Density red limit percentage  
- Persistense values  
- Persistence Infinite (enables/disables)  
- Enables/Disables Show Density graphics

Spectrogram
- Spectrogram blue limit percentage  
- Spectrogram red limit percentage  
- Time per division values  
- View choice

Last Modified:
20-sep-2016  Added new RTSA mode commands (9.50)

I/Q Analyzer (IQA) Mode Commands (A.09.xx and Greater Firmware Only)

In I/Q Analyzer (IQA) mode there are two main types of measurements: Spectrum and Waveform.
Here is the recommended procedure when querying trace-data or using marker-data using SCPI commands:

1. Set the Measurement to the desired type, if not already set:
   - For Spectrum data, use the :CONFigure:SPECTrum command to set the measurement to Spectrum.
   - For Waveform data, use the :CONFigure:WAVEform command to set the measurement to Waveform.

2. Change to Single acquisition mode by using INITiate:CONTInuous OFF, if the unit is not already in Single acquisition mode.

3. Execute one acquisition by using INITiate:IMMediate.

4. Perform the data-query (trace and/or marker) using the corresponding SCPI command.

Caution: Data integrity is not guaranteed if the above sequence of steps are not followed.

In this topic:
- **Frequency**
- **Gain/Atten**
- **Acquisition**
- **Scale / Units**
- **Average**
- **Video/ Resolution BW**
- **Measurement/Display**
- **Trace Type**
- **Alignments (InstAlign)**
- **Limit / Display Lines**
- **Markers**
- **Read / Save Data**
- **I/Q Capture**
- **Trigger Settings**
- **FFT** (Spectrum)
- **Sample**
- **Waveform**

**See Also**
- [Commands that are Common to All Modes](#)
- [Status Registers](#)

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>[:SENSe]:FREQuency:CENTer</td>
</tr>
<tr>
<td>Span</td>
<td>[:SENSe]:FREQuency:SPAN</td>
</tr>
</tbody>
</table>

**Gain/Atten**

- Attenuation value | [:SENSe]:POWer[:RF]:ATTenuation |
- Atten Auto/Manual  | [:SENSe]:POWer[:RF]:ATTenuation:AUTO |
- External gain      | [:SENSe]:POWer[:RF]:EXTGain |
- Preamp state       | [:SENSe]:POWer[:RF]:GAIN:AUTO |
- Preamp ON/OFF      | [:SENSe]:POWer[:RF]:GAIN[:STATe] |

**Acquisition**

- Enable/Disable acquisition mode | INITiate:CONTinuous |
- Restart trace averaging | INITiate:RESTart |

**Scale / Units**

- Auto Scale (IMAGinary) | TRACE:IMAGinary:Y:AUTO |
- Per division (IMAGinary) | TRACE:IMAGinary:Y:PDIVision |
<table>
<thead>
<tr>
<th>Mode</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference level (IMAGinary)</td>
<td>TRACe:IMAGinary:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (IMAGinary)</td>
<td>TRACe:IMAGinary:Y:RPOSition</td>
</tr>
<tr>
<td>Auto Scale (PHASE)</td>
<td>TRACe:PHASE:Y:AUTO</td>
</tr>
<tr>
<td>Per division (PHASE)</td>
<td>TRACe:PHASE:Y:PDIVision</td>
</tr>
<tr>
<td>Reference level (PHASE)</td>
<td>TRACe:PHASE:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (PHASE)</td>
<td>TRACe:PHASE:Y:RPOSition</td>
</tr>
<tr>
<td>Auto Scale (POLar)</td>
<td>TRACe:POLar:Y:AUTO</td>
</tr>
<tr>
<td>Reference level (POLar)</td>
<td>TRACe:POLar:Y:RLEVel</td>
</tr>
<tr>
<td>Auto Scale (REAL)</td>
<td>TRACe:REAL:Y:AUTO</td>
</tr>
<tr>
<td>Per division (REAL)</td>
<td>TRACe:REAL:Y:PDIVision</td>
</tr>
<tr>
<td>Reference level (REAL)</td>
<td>TRACe:REAL:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (REAL)</td>
<td>TRACe:REAL:Y:RPOSition</td>
</tr>
<tr>
<td>Log/Linear scale (SPECtrum)</td>
<td>TRACe:SPECtrum:AMPLitude:SCALE</td>
</tr>
<tr>
<td>Unit (SPECtrum)</td>
<td>TRACe:SPECtrum:AMPLitude:UNIT</td>
</tr>
<tr>
<td>Auto Scale (SPECtrum)</td>
<td>TRACe:SPECtrum:Y:AUTO</td>
</tr>
<tr>
<td>Per division (SPECtrum)</td>
<td>TRACe:SPECtrum:Y:PDIVision</td>
</tr>
<tr>
<td>Reference level (SPECtrum)</td>
<td>TRACe:SPECtrum:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (SPECtrum)</td>
<td>TRACe:SPECtrum:Y:RPOSition</td>
</tr>
<tr>
<td>Auto Scale (UPHase)</td>
<td>TRACe:UPHase:Y:AUTO</td>
</tr>
<tr>
<td>Per division (UPHase)</td>
<td>TRACe:UPHase:Y:PDIVision</td>
</tr>
<tr>
<td>Reference level (UPHase)</td>
<td>TRACe:UPHase:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (UPHase)</td>
<td>TRACe:UPHase:Y:RPOSition</td>
</tr>
<tr>
<td>Log/Linear scale (RF Envelope)</td>
<td>TRACe:WAVeform:AMPLitude:SCALE</td>
</tr>
<tr>
<td>Unit (RF Envelope)</td>
<td>TRACe:WAVeform:AMPLitude:UNIT</td>
</tr>
<tr>
<td>Auto Scale (RF Envelope)</td>
<td>TRACe:WAVeform:Y:AUTO</td>
</tr>
<tr>
<td>Per division (RF Envelope)</td>
<td>TRACe:WAVeform:Y:PDIVision</td>
</tr>
<tr>
<td>Reference level (RF Envelope)</td>
<td>TRACe:WAVeform:Y:RLEVel</td>
</tr>
<tr>
<td>Reference position (RF Envelope)</td>
<td>TRACe:WAVeform:Y:RPOSition</td>
</tr>
</tbody>
</table>

**Average**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average count</td>
</tr>
<tr>
<td>Averaging type</td>
</tr>
</tbody>
</table>
Restart trace averaging: INITiate:RESTart

**Video / Res Bandwidth**

- IF bandwidth (IF BW): [:SENSe]:DIF:BANDwidth
- Resolution bandwidth (Res BW): [:SENSe]:SPECtrum:BANDwidth[:RESolution]
- Resolution bandwidth mode (Res BW mode): [:SENSe]:SPECtrum:BANDwidth[:RESolution]:AUTO

**Measurement/Display**

- Configure query (Spectrum/Waveform): :CONFigure?
- Configure Spectrum command: :CONFigure:SPECtrum
- Configure Waveform command: :CONFigure:WAVeform
- Selected window: :DISPlay:WINDow[:SELect]
- Window display trace: (SPECtrum/WAVEform/PHASe/POLar/UPHase/REAL/IMAGinary/TIMesummary)
- Window State (Enable/Disable): :DISPlay:WINDow<n>:STATE

**Trace Type**

- Spectrum trace type: :TRACe:SPECtrum<n>:TYPE
- Waveform trace type: :TRACe:WAVeform<n>:TYPE

**Alignments (InstAlign)**

- Align all now (coupled to all individual alignments): [:SENSe]:ALIGNment:ALL:NOW
- Align all state (coupled to all individual alignments): [:SENSe]:ALIGNment:ALL[:STATe]
- Align now: [:SENSe]:ALIGNment:AMPLitude:NOW
- InstAlign state: [:SENSe]:ALIGNment:AMPLitude[:STATe]
- RF burst now: [:SENSe]:ALIGNment:BURSt:NOW
- RF burst state: [:SENSe]:ALIGNment:BURSt[:STATe]
- Channel equalization now: [:SENSe]:ALIGNment:CHEQ:NOW
- Channel equalization state: [:SENSe]:ALIGNment:CHEQ[:STATe]

**Limit / Display Lines**

- Display Line state: DISPlay:WINDow:TRACe:Y:DLIne[:STATe]
- Display line (IMAGinary): TRACe:IMAGinary:Y:DLIne
- Display line (PHASE): TRACe:PHASe:Y:DLIne
- Display line (REAL): TRACe:REAL:Y:DLIne
Display line (SPECTrum)  TRACe:SPECTrum:Y:DLINe
Display line (UPHase)  TRACe:UPHase:Y:DLINe
Display line (RF envelope)  TRACe:WAVeform:Y:DLINe

Markers
Markers all disabled (Spectrum)  CALCulate:SPECTrum:MARKer:AOFF
Continuous peak search (Spectrum) - (disable/enable)  CALCulate:SPECTrum:MARKer:CPSearch[:STATe]
Markers - Fixed delta reference marker state (Spectrum) - (disable/enable)  CALCulate:SPECTrum:MARKer:DREF:FIXed
Marker search - Peak excursion (Spectrum)  CALCulate:SPECTrum:MARKer:FUNCtion:PEXCursion
Marker search - Peak threshold (Spectrum)  CALCulate:SPECTrum:MARKer:FUNCtion:PTHReshold
Set Noise Marker and Band Power Marker (Spectrum)  CALCulate:SPECTrum:MARKer<n>:FUNCtion
Band power span  CALCulate:SPECTrum:MARKer:FUNCtion:BAND:SPAN
Band power mode  CALCulate:SPECTrum:MARKer:FUNCtion:BAND:SPAN:AUTO
Marker search - Maximum (Spectrum)  CALCulate:SPECTrum:MARKer<n>:FUNCtion:MAXimum
Marker search - Minimum (Spectrum)  CALCulate:SPECTrum:MARKer<n>:FUNCtion:MINimum
Marker search - Peak left (Spectrum)  CALCulate:SPECTrum:MARKer<n>:FUNCtion:PLEFt
Marker search - Peak right (Spectrum)  CALCulate:SPECTrum:MARKer<n>:FUNCtion:PRIGht
Read phase (Spectrum)  CALCulate:SPECTrum:MARKer<n>:PHASE?
Move marker to center frequency (Spectrum)  CALCulate:SPECTrum:MARKer<n>:SET:CENTer
Move marker to reference level (Spectrum)  CALCulate:SPECTrum:MARKer<n>:SET:RLEVel
Move marker to another specified trace (Spectrum)  CALCulate:SPECTrum:MARKer<n>:TRACe
Move/read marker X-axis position (Spectrum)  CALCulate:SPECTrum:MARKer<n>:X
Read marker Y-axis position (Spectrum)  CALCulate:SPECTrum:MARKer<n>:Y?
Marker State (Spectrum)  CALCulate:SPECTrum:MARKer<n>[STATe]
Markers - all disabled
(Waveform)

Coupled marker X value
(Waveform)

Markers - coupled (Waveform)

Continuous peak search
(Waveform) - (disable/enable)

Fixed delta / reference markers
(Waveform) - (disable/enable)

Marker search - Peak excursion
(Waveform)

Marker search - Peak threshold
(Waveform)

Set result trace to marker
(Waveform)

Set Noise Marker and Interval Span Marker (Waveform)

Interval span (Waveform)

Interval span mode (Waveform)

Marker search - Maximum
(Waveform)

Marker search - Minimum
(Waveform)

Marker search - Next Peak
(Waveform)

Move marker to another specified trace (Waveform)

Move/read marker X-axis position (Waveform)

Read marker Y-axis position (Waveform)

Marker State (Waveform)

Display marker table (Spectrum)

Display marker table (Waveform)

Read / Save
See Also: I/Q Capture
Store CSV file
Store MAT file
Store SDF file
Store TXT file
Reads the trace data (IMAGinary)
Reads the trace data (PHASe)
Reads the trace data (POLar)
Reads the trace data (REAL)
Reads the trace data (SPECtrum)
Reads the raw I/Q data (SPECtrum)
Returns the trace data (UPHase)
Reads the trace data (RF Envelope)
Reads the raw I/Q data (WAVEform)

I/Q Capture
See Also: Read / Save
Initiate I/Q capture
Device storage (i.e., INT, USB, or SD)
File count
File count mode
File name
File type (i.e., CSV, TXT, SDF, or MAT)
Start data capture
Stop data capture
Capture length
Capture time
Reads the I/Q data

Trigger Settings
Auto trigger time

COMMANDS

MMEMory:STORe:FDATA
MMEMory:STORe:MAT
MMEMory:STORe:SDF
MMEMory:STORe:TXT
TRACe:IMAGinary:DATA?
TRACe:PHASe:DATA?
TRACe:POLar:DATA?
TRACe:REAL:DATA?
TRACe:SPECtrum<n>:DATA?
TRACe:SPECtrum:RAWiq?
TRACe:UPHase:DATA?
TRACe:WAVEform<n>:DATA?
TRACe:WAVEform:RAWiq?
INITiate:IQCapture
MMEMory:STORe:IQCapture:DEVice
MMEMory:STORe:IQCapture:FCOunt
MMEMory:STORe:IQCapture:FCOunt:MULTiple
MMEMory:STORe:IQCapture:FNAMe
MMEMory:STORe:IQCapture:FTYPe
MMEMory:STORe:IQCapture:START
MMEMory:STORe:IQCapture:STOP
[:SENSe]:MEASure:CAPTure:LENGth
[:SENSe]:MEASure:CAPTure:TIME
TRACe:IQCapture:DATA?
TRIGger[:SEQuence]:ATRigger
Auto trigger ON/OFF
Trigger Delay (EXTernal)
Trigger Delay ON/OFF (EXTernal)
Trigger slope (Pos/Neg) - (EXTernal)
Trigger Delay (RF Burst)
Trigger Delay ON/OFF (RF Burst)
Glitch Mask (RF Burst)
Trigger Level (RF Burst)
Trigger slope (Pos/Neg) - (RF Burst)
Trigger type (Freerun/External/Video/RF Burst)
Trigger Delay (VIDeo)
Trigger Delay ON/OFF (VIDeo)
Trigger Level (VIDeo)
Trigger slope (Pos/Neg) - (VIDeo)
FFT (Spectrum)
FFT analysis length
FFT window length control (Enable/Disable)
FFT window
Sample
Sample period
Sample rate
Waveform
Start time
Stop time

FFT (Spectrum)

Sample

Waveform

Last Modified:
01nov2017 Added I/Q Mode (10.1x)
## USB Power Meter Mode Commands

- **Core USB Power Meter commands**
- **FOPS-(Option 208) unique commands**

### See Also
- [Commands that are Common to All Modes](#)
- [Pulse Measurements (Option 330) commands](#)
- [Status Registers](#)

### Core USB Power Meter commands

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set relative Power Meter measurements</td>
<td>CALCulate:RELative[:MAGNitude]:AUTO</td>
</tr>
<tr>
<td>Performs external power meter zeroing.</td>
<td>CALibration:ZERO:TYPE:EXT</td>
</tr>
<tr>
<td>Set Minimum scale value</td>
<td>DISPlay[:WINDow]:ANALog:LOWer</td>
</tr>
<tr>
<td>Set Maximum scale value</td>
<td>DISPlay[:WINDow]:ANALog:UPPer</td>
</tr>
<tr>
<td>Set PM resolution</td>
<td>DISPlay[:WINDow][:NUMeric]:RESolution</td>
</tr>
<tr>
<td>Query USB sensor for serial number &amp; model</td>
<td>:DISPlay:MODel:DATA?</td>
</tr>
<tr>
<td>Set number of sweep averages.</td>
<td>[:SENSe]:AVERage:COUNt</td>
</tr>
<tr>
<td>Set averaging mode</td>
<td>[:SENSe]:AVERage[:MODE]</td>
</tr>
<tr>
<td>Set Step Detection</td>
<td>[:SENSe]:AVERage:SDETect</td>
</tr>
<tr>
<td>Set Offset value.</td>
<td>[:SENSe]:CORRection:GAIN2[:INPut][:MAGNitude]</td>
</tr>
<tr>
<td>Set Offset ON</td>
<td>OFF state</td>
</tr>
<tr>
<td>Set frequency</td>
<td>[:SENSe]:FREQuency</td>
</tr>
<tr>
<td>Read measurement data</td>
<td>[:SENSe]:TRACe[:DATA]?</td>
</tr>
<tr>
<td>Set the minimum (lower) limit value.</td>
<td>[:SENSe]:TRACe:LIMit:LOWer</td>
</tr>
<tr>
<td>Set the lower ON</td>
<td>OFF State.</td>
</tr>
<tr>
<td>Set the maximum (upper) limit value.</td>
<td>[:SENSe]:TRACe:LIMit:UPPer</td>
</tr>
<tr>
<td>Set the upper ON</td>
<td>OFF State.</td>
</tr>
</tbody>
</table>
Set Source Enable: `SOURce:ENABle`
Set Source power level: `SOURce:POWer`
Set PM units: `UNIT:POWer`

**FOPS (Option 208) unique commands**
- Normalization
- Setup
- Display Annotation and Scaling
- Markers
- Trace Math

See Also: Example Program

**Perform Normalization**
There is NO unique FOPS command to MEASURE the source power. Instead, use the following method:
1. Prompt to connect the power sensor to port 1 RF Output reference plane.
2. Send `INIT:CONT 0`
3. Send `INITiate[:IMMediate]`
4. Send `SOURce:POWer:MEMorize` to store the data trace to memory.
5. Send `[:SENSe]:TRACe:MEASurement` to display your measurement choice.

**FOPS Setup**

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify swept frequency</td>
<td><code>[:SENSe]:SWEep:TYPE</code></td>
</tr>
<tr>
<td>Center frequency</td>
<td><code>SOURce:FREQuency:CENTer</code></td>
</tr>
<tr>
<td>Frequency span</td>
<td><code>SOURce:FREQuency:SPAN</code></td>
</tr>
<tr>
<td>Start frequency</td>
<td><code>SOURce:FREQuency:STARt</code></td>
</tr>
<tr>
<td>Stop frequency</td>
<td><code>SOURce:FREQuency:STOP</code></td>
</tr>
<tr>
<td>Offset frequency</td>
<td><code>SOURce:RECeiver:OFFSet</code></td>
</tr>
<tr>
<td>Number of points</td>
<td><code>[:SENSe]:SWEep:POINts</code></td>
</tr>
<tr>
<td>Frequency step size</td>
<td><code>[:SENSe]:FREQuency:STEP</code></td>
</tr>
<tr>
<td>Set dwell time</td>
<td><code>[:SENSe]:POINT:DWELI</code></td>
</tr>
<tr>
<td>Receiver sweep direction</td>
<td><code>[:SENSe]:SWEep:RX</code></td>
</tr>
<tr>
<td>Power sensor</td>
<td><code>[:SENSe]:TOL</code></td>
</tr>
</tbody>
</table>
Command by Mode

tolerance
Max number of PS readings [SENSe]:POINt:READ:MAX

Display Annotation and Scaling

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid ON/OFF</td>
<td>DISPlay:GRID</td>
</tr>
<tr>
<td>Method used to annotate frequency</td>
<td>DISPlay:ANNotation:FREQuency</td>
</tr>
<tr>
<td>Autoscale the trace</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:AUTO</td>
</tr>
<tr>
<td>Scaling - per division</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:PDIvision</td>
</tr>
<tr>
<td>Scaling - reference position</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:RPOSiQtion</td>
</tr>
<tr>
<td>Scaling - reference level</td>
<td>DISPlay:WINDow:TRACe{1:4}:Y[:SCALe]:RLEVEL</td>
</tr>
</tbody>
</table>

Markers

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a marker</td>
<td>CALCulate[:SElected]:MARKer:ACTivate</td>
</tr>
<tr>
<td>Marker ON/OFF</td>
<td>CALCulate[:SElected]:MARKer[:STATe]</td>
</tr>
<tr>
<td>Move a marker</td>
<td>CALCulate[:SElected]:MARKer:X</td>
</tr>
<tr>
<td>Read marker amplitude</td>
<td>CALCulate[:SElected]:MARKer:Y?</td>
</tr>
<tr>
<td>Set marker to Max</td>
<td>CALCulate[:SElected]:MARKer:FUNCTION:MAXimum</td>
</tr>
<tr>
<td>Set marker to Min</td>
<td>CALCulate[:SElected]:MARKer:FUNCTION:MINimum</td>
</tr>
<tr>
<td>Amp. markers ON/OFF</td>
<td>CALCulate[:SElected]:AMPLitude:MARKer:STATe</td>
</tr>
<tr>
<td>Amp. marker 1</td>
<td>CALCulate[:SElected]:AMPLitude:MARKer:Y1</td>
</tr>
<tr>
<td>Amp. marker 2</td>
<td>CALCulate[:SElected]:AMPLitude:MARKer:Y2</td>
</tr>
<tr>
<td>Delta Amp. markers ON/OFF</td>
<td>CALCulate[:SElected]:AMPLitude:MARKer:DELTa:STATe</td>
</tr>
</tbody>
</table>

Trace Math

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store a data trace to memory</td>
<td>CALCulate[:SElected]:MATH:MEMorize</td>
</tr>
<tr>
<td>Read data trace</td>
<td>CALCulate[:SElected]:TRACe:DATA</td>
</tr>
<tr>
<td>Read memory trace</td>
<td>CALCulate[:SElected]:FMEM:DATA?</td>
</tr>
</tbody>
</table>
Built-in Power Meter (CPM) Mode Commands

- **Frequency and Power**
- **Radio Standards**
- **Read Data**
- **Alignments (InstAlign)**
- **Display**
- **Limit Lines**

**See Also**
- [Commands that are Common to All Modes](#)
- [Status Registers](#)

## Frequency and Power

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set frequency</td>
<td>[:SENSe]:FREQuency</td>
</tr>
<tr>
<td>Step size for up/down keys</td>
<td>[:SENSe]:FREQuency:CENTer:STEP</td>
</tr>
<tr>
<td>Freq span</td>
<td>[:SENSe]:FREQuency:SPAN</td>
</tr>
<tr>
<td>Attenuation value</td>
<td>[:SENSe]:POWer[RF]:ATTenuation</td>
</tr>
</tbody>
</table>

## Radio Standard

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel number center</td>
<td>[:SENSe]:RADio:CHANnel:CENTer</td>
</tr>
<tr>
<td>Uplink or Downlink</td>
<td>[:SENSe]:RADio:CHANnel:DIREction</td>
</tr>
<tr>
<td>Channel step size</td>
<td>[:SENSe]:RADio:CHANnel:STEP</td>
</tr>
<tr>
<td>Select standard</td>
<td>[:SENSe]:RADio:STANdard[:SELect]</td>
</tr>
<tr>
<td>Freq or Chan</td>
<td>[:SENSe]:RADio:TEUNit</td>
</tr>
</tbody>
</table>

## Read Data

| Read measurement data     | [:SENSe]:TRACe[:DATA]?            |

## Alignments (InstAlign)

<table>
<thead>
<tr>
<th>Align all now (coupled to all individual alignments)</th>
<th>[:SENSe]:ALIGNment:ALL:NOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align all state (coupled to all individual alignments)</td>
<td>[:SENSe]:ALIGNment:ALL[:STATE]</td>
</tr>
</tbody>
</table>
Commands by Mode

Align now  [:SENSe]:ALIGNment:AMPLitude:NOW
InstAlign state  [:SENSe]:ALIGNment:AMPLitude[:STATe]

Display

Display units  [:SENSe]:AMPLitude:UNIT
Set Minimum scale value  DISPLAY[:WINDow]:ANALog:LOWer
Set Maximum scale value  DISPLAY[:WINDow]:ANALog:UPPer
Set PM resolution  DISPLAY[:WINDow][:NUMeric]:RESolution
Enable averaging  [:SENSe]:AVERage[:ENABLE]
Set number of sweep averages.  [:SENSe]:AVERage:COUNt
Make relative measurements  CALCulate:RELative[:MAGNitude]:AUTO
Set Offset value.  [:SENSe]:CORRection:GAIN2[:INPut][:MAGNitude]
Set Offset ON | OFF state  [:SENSe]:CORRection:GAIN2:STATe

Limit Lines

Set the minimum (lower) limit value.  [:SENSe]:TRACe:LIMit:LOWer
Set the lower ON | OFF State.  [:SENSe]:TRACe:LIMit:LOWer:STATe
Set the maximum (upper) limit value.  [:SENSe]:TRACe:LIMit:UPPer
Set the upper ON | OFF State.  [:SENSe]:TRACe:LIMit:UPPer:STATe

Last Modified:

1-Apr-2014 Added CPM commands (A.07.50)

VVM Mode Commands

See Also:
VVM Cable Trimming Example
Commands that are Common to All Modes

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create measurement</td>
<td>CALCulate:PARameter:DEFine</td>
</tr>
<tr>
<td>Averaging</td>
<td>[:SENSe]:AVERage:COUNt</td>
</tr>
<tr>
<td>Zeroing</td>
<td>[:SENSe]:CORRection:ZERO:STATe</td>
</tr>
<tr>
<td></td>
<td>[:SENSe]:CORRection:ZERO:REFerence</td>
</tr>
</tbody>
</table>
Set frequency [:SENSe]:FREQuency:CENTer
Read points (always 2) [:SENSe]:SWEep:POINts
Set power SOURce:POWer
Read data TRACe:DATA
IF Bandwidth [:SENSe]:BWID
Resolution DISPlay[:WINDow][:NUMeric]:RESolution

Pulse Measurements (Option 330) Commands

The following commands are part of the USB Power Meter mode.

- Select a Measurement / Trace
- Time / Frequency
- Average and Bandwidth Video
- Scale
- Display
- Trigger
- Markers
- Limits
- Read / Save Data

See Also
- USB PM Mode Commands
- Commands that are Common to All Modes
- Status Registers

Select a Measurement / Trace

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set measurement.</td>
<td>CALCulate:FEED:MODE</td>
</tr>
</tbody>
</table>

Time / Frequency

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center time of zoom window</td>
<td>CALCulate[:SELected]:TIME:AUX:CENTer</td>
</tr>
<tr>
<td>Time/div of zoom window</td>
<td>CALCulate[:SELected]:TIME:AUX:PDIVision</td>
</tr>
<tr>
<td>Center time of trace graph</td>
<td>CALCulate[:SELected]:TIME:CENTer</td>
</tr>
<tr>
<td>Span time of trace graph</td>
<td>CALCulate[:SELected]:TIME:LENGTH</td>
</tr>
<tr>
<td>Description</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Time/div of trace graph</td>
<td>CALCulate[:SELECTed]:TIME:PDIVision</td>
</tr>
<tr>
<td>Start time of trace graph</td>
<td>CALCulate[:SELECTed]:TIME:STARt</td>
</tr>
<tr>
<td>Frequency of meas</td>
<td>[:SENSe]:FREQuency</td>
</tr>
<tr>
<td>Number of points</td>
<td>[:SENSe]:RESolution</td>
</tr>
</tbody>
</table>

### Average and Bandwidth Video

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of averages</td>
<td>[:SENSe]:AVERage:COUNT</td>
</tr>
<tr>
<td>Averaging Auto, Man, Off</td>
<td>[:SENSe]:AVERage[:MODE]</td>
</tr>
<tr>
<td>Step detection mode</td>
<td>[:SENSe]:AVERage:SDETECT</td>
</tr>
<tr>
<td>Set and query the IF bandwidth</td>
<td>[:SENSe]:BWID</td>
</tr>
<tr>
<td>Video bandwidth</td>
<td>[:SENSe]:BWIDth:VIDeo</td>
</tr>
</tbody>
</table>

#### Scale

**Meter** = Meter-style only; **TG** = Trace Graph only; **Both** = Meter and Trace Graph

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meter - Relative ON/OFF</td>
<td>CALCulate:RELative[:MAGNitude]:AUTO</td>
</tr>
<tr>
<td>Meter - Min Scale</td>
<td>DISPLAY:WINDow:ANALog:LOWER</td>
</tr>
<tr>
<td>Meter - Max Scale</td>
<td>DISPLAY:WINDow:ANALog:UPPer</td>
</tr>
<tr>
<td>Meter - Resolution</td>
<td>DISPLAY:WINDow[:NUMeric]:RESolution</td>
</tr>
<tr>
<td>TG - Autoscale</td>
<td>None</td>
</tr>
<tr>
<td>TG - Scale</td>
<td>None</td>
</tr>
<tr>
<td>TG - Ref Lv</td>
<td>None</td>
</tr>
<tr>
<td>TG - Ref Pos</td>
<td>None</td>
</tr>
<tr>
<td>Both - Scale Offset ON/OFF</td>
<td>[:SENSe]:CORRection:GAIN2:STATe</td>
</tr>
<tr>
<td>Both - Scale Offset value</td>
<td>[:SENSe]:CORRection:GAIN2[:INPut][:MAGNitude]</td>
</tr>
</tbody>
</table>

#### Display

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid ON/OFF</td>
<td>DISPLAY:GRID</td>
</tr>
<tr>
<td>Marker Table ON/OFF</td>
<td>DISPLAY:TABLE:MARKer</td>
</tr>
<tr>
<td>Auto Analysis ON/OFF</td>
<td>DISPLAY:TABLE:RESults</td>
</tr>
<tr>
<td>Description</td>
<td>Command</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>New Printed Documentation Output 19dec2018FINAL</td>
<td>DISPlay:TABLE:RESults:DATA</td>
</tr>
<tr>
<td>Zoom window ON/OFF</td>
<td>DISPlay:WINDow:ZOOM</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td><strong>Command</strong></td>
</tr>
<tr>
<td>Internal, External, Freerun</td>
<td>TRIGger:SOUrce</td>
</tr>
<tr>
<td>Trigger delay value</td>
<td>TRIGger:DELay</td>
</tr>
<tr>
<td>Trigger level value</td>
<td>TRIGger:LEVel</td>
</tr>
<tr>
<td>Auto or Manual level</td>
<td>TRIGger:LEVel:AUTO</td>
</tr>
<tr>
<td>Pos or Neg edge</td>
<td>TRIGger:SLOPe</td>
</tr>
<tr>
<td><strong>Markers</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td><strong>Command</strong></td>
</tr>
<tr>
<td>Select a marker</td>
<td>CALCulate[:SELected]:MARKer:ACTivate</td>
</tr>
<tr>
<td>Marker ON/OFF</td>
<td>CALCulate[:SELected]:MARKer[:STATe]</td>
</tr>
<tr>
<td>Move a marker</td>
<td>CALCulate[:SELected]:MARKer:X</td>
</tr>
<tr>
<td>Read marker amplitude</td>
<td>CALCulate[:SELected]:MARKer:Y?</td>
</tr>
<tr>
<td>Set markers to Falltime</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:FALLtime</td>
</tr>
<tr>
<td>Set markers to Risetime</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:RISetime</td>
</tr>
<tr>
<td>Set marker to Max</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:MAXimum</td>
</tr>
<tr>
<td>Set marker to Min</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:MINimum</td>
</tr>
<tr>
<td>Peak Excursion value</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:PEXCursion</td>
</tr>
<tr>
<td>Peak Threshold value</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:PTHReshold</td>
</tr>
<tr>
<td>Find Next Peak</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:PNEXT</td>
</tr>
<tr>
<td>Find Target value</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:TARGet</td>
</tr>
<tr>
<td>Marker search - Wrap/No Wrap</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:TDIREction</td>
</tr>
<tr>
<td>Set marker tracking</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:TRACking</td>
</tr>
<tr>
<td>Search zoom window or primary trace.</td>
<td>CALCulate[:SELected]:MARKer:FUNCTION:ZONE</td>
</tr>
<tr>
<td>Delta Amp. markers ON/OFF</td>
<td>CALCulate[:SELected]:AMPLitude:MARKer:DELTa:STATe</td>
</tr>
<tr>
<td>Amp. markers ON/OFF</td>
<td>CALCulate[:SELected]:AMPLitude:MARKer:STATe</td>
</tr>
<tr>
<td>Amp. marker 1</td>
<td>CALCulate[:SELected]:AMPLitude:MARKer:Y1</td>
</tr>
</tbody>
</table>
Amp. marker 2

CALCulate[:SELeCted]:AMPLitude:MARKer:Y2

Pulse top

[SENSe]:TRACe:MEASurement:REFerence

**Read / Save Data**

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read trace graph data.</td>
<td>CALCulate[:SELeCted]:TRACe:DATA</td>
</tr>
<tr>
<td>Read Meter-style data</td>
<td>[:SENSe]:TRACe[DATA]?</td>
</tr>
<tr>
<td>Save to *.csv file</td>
<td>MMEMory:STORE:FDATa</td>
</tr>
</tbody>
</table>

**Limits (Meter-style ONLY)**

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower limit value</td>
<td>[:SENSe]:TRACe:LIMit:LOWer</td>
</tr>
<tr>
<td>Lower limit state</td>
<td>[:SENSe]:TRACe:LIMit:LOWer:STATe</td>
</tr>
<tr>
<td>Upper limit value</td>
<td>[:SENSe]:TRACe:LIMit:UPPer</td>
</tr>
<tr>
<td>Upper limit state</td>
<td>[:SENSe]:TRACe:LIMit:UPPer:STATe</td>
</tr>
</tbody>
</table>

**ERTA Mode Commands**

ONLY ERTA specific commands are shown here.

Use **SA Mode Commands** for all other relevant settings not listed here. For example, Frequency range and Tracking Offset commands.

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set and query the partner network identity.</td>
<td>[:SENSe]:MEASurement:ERTA:PNID</td>
</tr>
<tr>
<td>Verify the identified partner is ERTA capable.</td>
<td>[:SENSe]:MEASurement:ERTA:PVERify?</td>
</tr>
<tr>
<td>Set ERTA stimulus-response role.</td>
<td>[:SENSe]:MEASurement:ERTA:ROLE?</td>
</tr>
<tr>
<td>Set and read Partnership status.</td>
<td>[:SENSe]:MEASurement:ERTA:PSTatus</td>
</tr>
<tr>
<td>Trace Receiver Input, valid during ERTA partnership</td>
<td>TRACe:ERTA:RINPut</td>
</tr>
</tbody>
</table>

**See Also**

- Commands that are Common to All Modes
- Status Registers
Channel Scanner Mode (Option 312) Commands - Requires SA

In this topic:

- **Data**
- **Display**
- **Edit List**
- **Range**
- **Field Strength (Corrections)**
- **Receiver (or Only one) Antenna/Cables**
- **Log and Recording**
- **File Commands**
- **Search Channels**
- **Alignments (InstAlign)**
- **Channel Power**
- **Units**
- **Sweep**
- **SA Listen**
- **Related MMEMory commands**

**See Also**

- [Commands that are Common to All Modes](#)
- [Correction Methods Explained](#)
- [New Programming Commands](#)
- [Status Registers](#)

**Data**

- Returns a set of values: :CHSCanner:DATA?
- Sets the folder path to Default or System: :CHSCanner:USER:FOLDER

**Display**

- Set freq and pow: :CHSCanner:DISPLAY:SORT
- Set up and down sort order: :CHSCanner:DISPLAY:SORT:ORDer
- Set display window trace Y Pow division: :CHSCanner:DISPLAY:WINDow:TRACe:Y[:SCALe]:PDIVsion
- Set display window trace Y Ref level: :CHSCanner:DISPLAY:WINDow:TRACe:Y[:SCALe]:RLEVel

**Edit List**

- Returns a set of values: :CHSCanner:EDIT:LIST?
- Add comma separated list item: :CHSCanner:EDIT:LIST:ADD
- Clears all items from list: :CHSCanner:EDIT:LIST:CLEar

**Range**

99
Commands by Mode

Reads or writes items in Range mode
:CHS:EDIT:RANGe:COUNt

Set integration bandwidth
:CHS:EDIT:RANGe:IBW

Field Strength (Corrections)
All corrections ON [:SENSe]:AMPLitude:CORRections[:STATe]

Receiver (or ONLY one) Antenna/Cables

Load Antenna file
MMEMory:LOAD:ANTenna

Store Antenna file
MMEMory:STORe:ANTenna

Load Cable file
MMEMory:LOAD:CABLE

Store Cable file
MMEMory:STORe:CABLE

Antenna corrections ON [:SENSe]:AMPLitude:CORRections:ANTenna[:STATe]

Cable corrections ON [:SENSe]:AMPLitude:CORRections:CABLE[:STATe]

Clear Antenna correction values [:SENSe]:AMPLitude:CORRections:ANTenna:DEFault

Clear Cable correction values [:SENSe]:AMPLitude:CORRections:CABLE:DEFault

Display corrections window [:SENSe]:AMPLitude:CORRections:VIEW

Log and Recording

Begin recording results :CHSC:LOG:ACTion:RECord

Stop recording :CHSC:LOG:ACTion:STOP

Read and writes the distance interval
:CHSC:LOG:CONFigure:INTerval:DISTance

Enable or disable the measurement interval
:CHSC:LOG:CONFigure:INTerval:STATe

Read and write the time interval
:CHSC:LOG:CONFigure:INTerval:TIMe

Set type of interval
:CHSC:LOG:CONFigure:INTerval:TYPE

Enable or disable the log file auto-save
:CHSC:LOG:CONFigure:SAVe:AUTo

Query the state :CHSC:LOG[:STATe]?

File Commands

Sets the user folder path to default or system
:CHSC:USER:FOLDer

Search Channels

Set the number of channels to scan
:CHSC:SEARch:COUNt

Set top or bottom number of
:CHSC:SEARch:TYPE
channels

**Alignments (InstAlign)**

Align all now (coupled to all individual alignments)  
[:SENSe]:ALIGNment:ALL:NOW

Align all state (coupled to all individual alignments)  
[:SENSe]:ALIGNment:ALL[:STATe]

Align now  
[:SENSe]:ALIGNment:AMPLitude:NOW

InstAlign state  
[:SENSe]:ALIGNment:AMPLitude[:STATe]

**Channel Power**

Set the active channel  
:CHSCanner[:SENSe]:CHANnel:SELection

Set the RF attenuator value manually  
:CHSCanner[:SENSe]:POWer[:RF]:ATTenuation

Set the RF attenuator to Auto  
:CHSCanner[:SENSe]:POWer[:RF]:ATTenuation:AUTO

Set the external Gain  
:CHSCanner[:SENSe]:POWer[:RF]:EXTGain

Enable or disable the Gain state  
:CHSCanner[:SENSe]:POWer[:RF]:GAIN[:STATe]

**Units**

Set Units  
[:SENSe]:AMPLitude:UNIT

**Sweep**

Set the averaging count  
:CHSCanner:SWEep:AVERage:COUNt

Enable or Disable the averaging state  
:CHSCanner:SWEep:AVERage[:STATe]

Set the displayed sweep type  
:CHSCanner:SWEep:DISPlay:TYPE

Set the sweep mode  
:CHSCanner:SWEep:MODE

**SA Listen**

Set SA Listen D type  
:CHSCanner[:SENSe]:SAListen:DTYPE

Set SA Listen L Time value  
:CHSCanner[:SENSe]:SAListen:LTIMe

Pause data recording  
:CHSCanner[:SENSe]:SAListen:PAUSE

Resume data recording  
:CHSCanner[:SENSe]:SAListen:RESume

**MMEMory commands - related to Channel Scanner**

Save a CSV formatted file to Channel Scanner folder  
:MMEMory:STORe:LOG:CSV

Save a KML formatted file to Channel Scanner folder  
:MMEMory:STORe:LOG:KML

Load a *.csv or *.kml log file  
:MMEMory:LOAD:LOG

Load a *.csv custom list file  
:MMEMory:LOAD:LIST
Save a *csv custom list file :MMEMory:STATe:STORe:LIST

LTE FDD Mode (Option 370) Commands - Requires SA and GPS

In this topic:
- Data
- Display
- Favorites List
- Frequency / Carrier
- Scale/Units
- Alignments (InstAlign)
- Power
- Trigger Settings
- Record Playback Actions
- Record Playback Configurations
- Sweep
- Related MMEMory commands

See Also
- Commands that are Common to All Modes
- Correction Methods Explained
- New Programming Commands
- Status Registers

Data
Returns a set of values with GPS information.
LTEFdd:DATA:GPS?

Returns a set of values without GPS information.
LTEFdd:DATA?

Display
Sets the trace data type (RSRP | RSRQ |RSSI | PSS | SSS | SINR )
DISPlay:LTEFdd:TRACe:DATA

Set the selected window to be displayed.
DISPlay:LTEFdd:WINDow[:SELect]

Set and query the component carrier (cc).
DISPlay:LTEFdd:WINDow[1]|2|3|4:CARRier

Set the window trace type (Table, Bar Chart, Spectrum, Strip Chart).
DISPlay:LTEFdd:WINDow[1]|2|3|4:DATA

Sets the order of the data sorting (AUTO | UP | DOWN)
Sets the type of data being sorted (RSRP | RSRQ | RSSI | PSS | SSS | SINR)

**Display:** `LTEFdd:WINDow[1]|2|3|4:SORT:ORDer`

Set the selected window state.

**Display:** `LTEFdd:WINDow[1]|2|3|4:STATe`

**Favorites List**

Set the current channel and band to a favorites setup

**[:SENSe]:LTEFdd:CCARrier:LIST[1]|2|3|4|5:ADD**

Query the band in the current selected favorites setup

**[:SENSe]:LTEFdd:CCARrier:LIST[1]|2|3|4|5:BAND?**

Query the channel in the current selected favorites setup

**[:SENSe]:LTEFdd:CCARrier:LIST[1]|2|3|4|5:CHANnel?**

Remove the current channel and band favorites setup

**[:SENSe]:LTEFdd:CCARrier:LIST[1]|2|3|4|5:REMove**

Apply a favorites setup—band and channel—to a measurement


**Frequency / Carrier**

Set and query the frequency error threshold

**[:SENSe]:LTEFdd:FERRor:THReshold**

Set and query the extended frequency lock

**[:SENSe]:LTEFdd:FLRange:EXTended**

Set and query the component carrier band

**[:SENSe]:LTEFdd:CCARrier[1]|2|3|4|5:BAND**

Set and query the component carrier channel

**[:SENSe]:LTEFdd:CCARrier[1]|2|3|4|5:CHANnel**

Set and query the selected carrier

**[:SENSe]:LTEFdd:CCARrier[1]|2|3|4|5:ENABle**

Set and query the center frequency of each component carrier (CC)

**[:SENSe]:LTEFdd:CCARrier[1]|2|3|4|5:FREQuency:CENTer**

Set and query the component carrier mode (CHANnel | FREQuency)

**[:SENSe]:LTEFdd:CCARrier[1]|2|3|4|5:MODE**

**Scale / Units**

Auto Scale the bar chart scan data

**Display:** `LTEFdd:BCHart:Y[:SCALe]:AUTO`

Set and query the per division value of the Bar Graph data

**Display:** `LTEFdd:BCHart:Y[:SCALe]:PDIVision`

Set and query the Bar Chart reference value

**Display:** `LTEFdd:BCHart:Y[:SCALe]:RLEVeL`

Auto Scale the Strip chart scan data

**Display:** `LTEFdd:SCHart:Y[:SCALe]:AUTO`

Set and query the per division

**Display:** `LTEFdd:SCHart:Y[:SCALe]:PDIVision`
value of the Strip Chart data

Set and query the Strip Chart reference value

Auto Scale the Spectrum chart scan data

Set and query the per division value of the Spectrum data

Set and query the Spectrum reference value

Set and query the reference position of the Spectrum trace

Alignments (InstAlign)

Align all now (coupled to all individual alignments)

Align all state (coupled to all individual alignments)

Align now

InstAlign state

Power

Set the RF attenuator value manually

Set the RF attenuator to Auto

Set the external Gain

Enable or disable the Gain state

Set the absolute power level

Set the red bar chart limit

Set the blue bar chart limit

Trigger Settings

Auto trigger time

Auto trigger ON/OFF

Trigger Slope (Pos/Neg)

Trigger Delay

Trigger Delay ON/OFF

Trigger Type (Ext/Freerun)

Record/Playback Actions

Pause
Play

Sets the position number of component carrier manually, when paused.

Sets the position number of component carrier automatically, when paused.

Record

Save (.kml or .csv)

Stop

Record/Playback Configuration

Sets the device type storage location (INTernal | USB | SD)

Enables/Disables overwrite data filename

Sets the data storage type (KML | CSV)

Set the recording distance interval

Enables/Disables the measurement interval requirement for saving records

Set and query time interval in seconds

Sets the measurement interval for saving records (TIME | DISTance)

Sweep

Enables/disables continuous sweep

Restarts the trace averaging

MMEMory commands - related to LTE FDD

Save a file to the default folder

Load a *.csv or *.kml from an OTA recorder folder

5GTF Mode (Option 377) Commands - Requires SA and GPS

In this topic:
Commands by Mode

- **Data**
- **Display**
- **Frequency / Carrier**
- **Scale / Units**
- **Alignments (InstAlign)**
- **Power**
- **Trigger Settings**
- **Record Playback Actions**
- **Record Playback Configurations**
- **Sweep**
- Related **MMEbory commands**

**See Also**
- Commands that are Common to All Modes
- Correction Methods Explained
- New Programming Commands
- Status Registers

**Data**

- Returns a set of values with GPS information.
  
  V5G:DATA:GPS?

- Returns a set of values without GPS information.
  
  V5G:DATA?

- Sets the trace data type (PSS | SSS | POW)
  
  DISPlay:V5G:TRACe:DATA

- Set and query the component carrier (cc).
  

- Set the window trace type (Table, Bar Chart, Spectrum, Strip Chart).
  

- Sets the order of the data sorting (AUTO | UP | DOWN)
  

- Sets the type of data being sorted (PSS | SSS | POW)
  

**Display**

- Set the selected window to be displayed.
  
  DISPlay:V5G:WINDow:[SELeCT]

- Set the selected window state.
  

**Frequency / Carrier**

- Set and query the sync correlation threshold percentage
  
  [:SENSe]:V5G:SCORr:THReshold

- Set and query the selected carrier
  
  [:SENSe]:V5G:CCARrier[1]|2|3|4|5|6|7|8:ENABLE
Set and query the center frequency of each component carrier (CC)

[:SENSe]:V5G:CCARrier[1]|2|3|4|5|6|7|8:FREQuency:CENTer

**Scale / Units**

Auto Scale the bar chart scan data

DISPLAY:V5G:BCHart:Y[:SCALe]:AUTO

Set and query the per division value of the Bar Graph data

DISPLAY:V5G:BCHart:Y[:SCALe]:PDIVision

Set and query the Bar Chart reference value

DISPLAY:V5G:BCHart:Y[:SCALe]:RLEVel

Auto Scale the Strip chart scan data

DISPLAY:V5G:SCHart:Y[:SCALe]:AUTO

Set and query the per division value of the Strip Chart data

DISPLAY:V5G:SCHart:Y[:SCALe]:PDIVision

Set and query the Strip Chart reference value

DISPLAY:V5G:SCHart:Y[:SCALe]:RLEVel

Auto Scale the Spectrum chart scan data

DISPLAY:V5G:SPECtrum:Y[:SCALe]:AUTO

Set and query the per division value of the Spectrum data

DISPLAY:V5G:SPECtrum:Y[:SCALe]:PDIVision

Set and query the per division reference value of the Spectrum trace

DISPLAY:V5G:SPECtrum:Y[:SCALe]:RLEVel

Set and query the reference position of the Spectrum trace

DISPLAY:V5G:SPECtrum:Y[:SCALe]:RPOSition

**Alignments (InstAlign)**

Align all now (coupled to all individual alignments)

[:SENSe]:ALIGNment:ALL:NOW

Align all state (coupled to all individual alignments)

[:SENSe]:ALIGNment:ALL[:STATe]

Align now

[:SENSe]:ALIGNment:AMPLitude:NOW

InstAlign state

[:SENSe]:ALIGNment:AMPLitude[:STATe]

**Power**

Set the RF attenuator value manually

[:SENSe]:POWer[:RF]:ATTenuation

Set the RF attenuator to Auto

[:SENSe]:POWer[:RF]:ATTenuation:AUTO

Set the external Gain

[:SENSe]:POWer[:RF]:EXTGain

Enable or disable the Gain state

[:SENSe]:POWer[:RF]:GAIN[:STATe]

Set the absolute power level

[:SENSe]:POWer[:RF]:RLEVel
Set the red bar chart limit [:SENSe]:POWer[:RF]:RPLevel
Set the blue bar chart limit [:SENSe]:POWer[:RF]:BPLevel

**Trigger Settings**

Auto trigger time TRIGger[:SEQuence]:ATRigger
Auto trigger ON/OFF TRIGger[:SEQuence]:ATRigger:STATe
Trigger Type (Ext/Freerun) TRIGger[:SEQuence]:SOURce
Trigger Slope (Pos/Neg) TRIGger[:SEQuence]:EXTernal:SLOPe
Trigger Delay TRIGger[:SEQuence]:EXTernal:DELay
Trigger Delay ON/OFF TRIGger[:SEQuence]:EXTernal:DELay:STATe

**Record/Playback Actions**

Pause RECPPlayback:ACTion:PAUSe
Play RECPPlayback:ACTion:PLAY
Sets the position number of component carrier manually, when paused. RECPPlayback:ACTion:POSition
Sets the position number of component carrier automatically, when paused. RECPPlayback:ACTion:POSition:AUTO
Record RECPPlayback:ACTion:RECord
Save (.kml or .csv) RECPPlayback:ACTion:SAVE
Stop RECPPlayback:ACTion:STOP

**Record/Playback Configuration**

Sets the device type storage location (INTernal | USB | SD) RECPPlayback:CONFig:FILE:DEVice
Enables/Disables overwrite data filename RECPPlayback:CONFig:FILE:OWRite
Sets the data storage type (KML | CSV) RECPPlayback:CONFig:FILE:TYPE
Set the recording distance interval RECPPlayback:CONFig:INTerval:DISTance
Enables/Disables the measurement interval requirement for saving records RECPPlayback:CONFig:INTerval:STATe
Set and query time interval in seconds RECPPlayback:CONFig:INTerval:TIME
Sets the measurement interval for saving records (TIME | DISTance RECPPlayback:CONFig:INTerval:TYPE

**Sweep**
Enables/disables continuous sweep:

:INITiate:CONTinuous

Enable or Disable the averaging state:

:INITiate:RESTart

**MMEMory commands - related to 5GTF**

Save a file to the default folder:

:MMEMory:STORe:FDATa

Load a *.csv or *.kml from an OTA recorder folder:

:MMEMory:LOAD:OTA:LOG

---

**Noise Figure (NF) Mode (Option 356) Commands (A.10.3x and Greater Firmware Only)**

In NF (Noise Figure) mode there are four types of measurements: Noise Figure, Noise Factor, Gain, Noise Temperature, and Y-Factor. Here is an example procedure for setting up a noise figure measurement using SCPI commands:

1. Set up the noise source and ENR table, using: CORR:ENR:MOD and set to TABLe or SPOT
2. And then
   - For SPOT commands in the Noise Source / ENR table below (*.enr)
3. Set the noise bandwidth, using the NBANDwidth command
4. Set the number of points, using SWE:POIN
5. Enter a DUT setup type, using the DUT commands table
6. Set the frequency range, using Frequency commands table
7. Setup Integration, using the Integration commands table
8. Set the uncertainty contributions, using Uncertainty commands table
9. If you haven't run a receiver calibration, then perform a Receiver Cal, using the Calibration commands table
10. Run a user calibration, using Calibration commands table
11. Set the type of noise figure measurement (Noise Figure (NFIG), Noise Factor (NFAC), Gain (GAIN), Noise Temperature (NTEM), or Y-Factor (YFAC)), using the CALCulate:PARameter:DEFine

For more information on the calibration and DUT measurement setup user interface, refer to the User's Guide (N9927-90001).

In this topic:

- Frequency
Commands by Mode

- DUT
- Sweep
- Scale / Units
- Traces
- Integration
- Noise / ENR Data
- DUT Data
- Preamplifier Data
- Uncertainty Data
- Resolution Bandwidth
- Alignments (InstAlign)
- Limit Lines
- Markers
- Calibration
- Read / Save Data

See Also
- Commands that are Common to All Modes
- Status Registers

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center freq</td>
<td>[:SENSe]:FREQuency:CENTer</td>
</tr>
<tr>
<td>Frequency context</td>
<td>[:SENSe]:FREQuency:CONText</td>
</tr>
<tr>
<td>Read X-axis values</td>
<td>[:SENSe]:FREQuency:DATA?</td>
</tr>
<tr>
<td>IF (DUT Out) start frequency</td>
<td>[:SENSe]:FREQuency:IF:STARt?</td>
</tr>
<tr>
<td>LO</td>
<td>[:SENSe]:FREQuency:LO</td>
</tr>
<tr>
<td>RF (DUT In) Start Frequency</td>
<td>[:SENSe]:FREQuency:RF:STARt?</td>
</tr>
<tr>
<td>Freq span</td>
<td>[:SENSe]:FREQuency:SPAN</td>
</tr>
<tr>
<td>Start</td>
<td>[:SENSe]:FREQuency:STARt</td>
</tr>
<tr>
<td>Stop</td>
<td>[:SENSe]:FREQuency:STOP</td>
</tr>
</tbody>
</table>

DUT

- After DUT loss (double)   [:SENSe]:CORRection:LOSS:AFTer[:VALue]
- Before DUT loss (double)  [:SENSe]:CORRection:LOSS:BEFore[:VALue]
- After loss enabled (bool) [:SENSe]:CORRection:LOSS:AFTer:ENABled
- Before loss enabled (bool) [:SENSe]:CORRection:LOSS:BEFore:ENABLed
- Before DUT temp (double)  [:SENSe]:CORRection:BEFore:TEMPerature
After DUT temp (double) [:SENSe]:CORRection:AFTer:TEMPerature
Set DUT type [:SENSe]:DUT[:TYPe]
Set Sideband [:SENSe]:DUT:SIDEband
Sweep
Sweep points [:SENSe]:SWEep:POINts
Scale / Units
Scaling - auto DISPlay:WINDow:TRACe<n>:Y[:SCALe]:AUTO
Scaling - Set bottom of scale DISPlay:WINDow:TRACe<n>:Y[:SCALe]:BOTTom
Scaling - Set per division DISPlay:WINDow:TRACe<n>:Y[:SCALe]:PDIVision
Scaling - Set reference level DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RLEVel
Scaling - Set reference position DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RPOSition
Scaling - Set top of scale DISPlay:WINDow:TRACe<n>:Y[:SCALe]:TOP
Traces
Create measurement CALCulate:PARameter:DEFine
Set and read number of traces CALCulate:PARameter:COUNt
Select Measurement CALCulate:PARameter:SELect
Multi-trace Configurations DISPlay:WINDow:SPLit
View Memory trace DISPlay:WINDow:TRACe<n>:MEMory:STATe
View Data trace DISPlay:WINDow:TRACe<n>:STATe
Stores current trace points DISPlay:WINDow:TRACe<n>:STORe
Integration
Jitter goal [:SENSe]:INTegration:JITTer
Maximum time per point [:SENSe]:INTegration:MAXTime
Mode [:SENSe]:INTegration:MODe
Time per point [:SENSe]:INTegration:TIme
Jitter warning [:SENSe]:INTegration:WARNing
Jitter goal exceeded warning (query) TRACe<n>:JITTer:WARNing:DATA?
Jitter goal exceeded memory warning (query) TRACe<n>:MEMory:JITTer:WARNing:DATA?
Noise / ENR Data
ENR Extrapolation state (query only) [:SENSe]:CORRection:ENR:EXTRapolated?
Spot ENR uncertainty coverage [:SENSe]:CORRection:ENR:SPOT:COVerage
Spot ENR gamma distribution  
ENR mode  
Spot ENR off gamma  
Spot ENR on gamma  
Spot ENR gamma specify style  
Spot ENR value  
Spot ENR uncertainty  
T Cold  
Recall ENR data  
Save ENR data  

**DUT Data**

Spot DUT distribution  
Spot DUT in gamma  
DUT mode  
Spot DUT out gamma  
Spot DUT specify style  
Import DUT match data  
Recall DUT data  
Save DUT data  

**Preamplifier Data**

Spot preamp distribution  
Spot preamp in gamma  
Preamp mode  
Spot preamp out gamma  
Spot preamp specify style  
Import preamplifier match data  
Recall preamplifier data  
Save preamplifier data  

**Uncertainty**

Error bars (Enable/Disable)  
Apply calibration uncertainty
Uncertainty coverage[:SENSe]:CORRection:UNCertainty:COVerage
Apply noise source uncertainty[:SENSe]:CORRection:UNCertainty:ENR
Apply jitter uncertainty[:SENSe]:CORRection:UNCertainty:JITTER
Apply mismatch uncertainty[:SENSe]:CORRection:UNCertainty:MISMatch
Uncertainty questionable state (query only):[:SENSe]:CORRection:UNCertainty:QUEStionable?

Resolution Bandwidth
Manual resolution bandwidth value[:SENSe]:NBANdwidth

Alignments
Align now[:SENSe]:ALIGnment:AMPLitude:NOW
InstAlign state[:SENSe]:ALIGnment:AMPLitude[:STATe]

Limit / Display Lines
Limit line beepCALCulate[:SELected]:LIMit:SOUNNd

Create limit linesCALCulate[:SELected]:LIMit:LLData
Limit line testing stateCALCulate:LIMit[:STATe]
Limit line testing annotationCALCulate:LIMit:WARN
Display Line level settingDISPlay:WINDow:TRACe:Y:DLINe
Display Line stateDISPlay:WINDow:TRACe:Y:DLINe:STATe
Read Pass / FailSTATus:QUEStionable:LIMit:CONDition?
Build Limit from TraceNone

Markers
Select a markerCALCulate[:SELected]:MARKer:ACTivate
Markers - all offCALCulate[:SELected]:MARKer:AOFF
Marker search - MaxCALCulate[:SELected]:MARKer:FUNCtion:MAXimum
Marker search - MinCALCulate[:SELected]:MARKer:FUNCtion:MINimum
Marker search - Peak excursionCALCulate[:SELected]:MARKer:FUNCtion:PEXCursion
Marker search - Peak left  
CALCulate[:SELECTed]:MARKer:FUNCTION:PLEFt

Marker search - Peak Next  
CALCulate[:SELECTed]:MARKer:FUNCTION:PNEXT

Marker search - Peak right  
CALCulate[:SELECTed]:MARKer:FUNCTION:PRIGHT

Marker search - Peak threshold  
CALCulate[:SELECTed]:MARKer:FUNCTION:PTHreshold

Marker -> Setting  
CALCulate[:SELECTed]:MARKer:SET

Move marker to center freq  
CALCulate[:SELECTed]:MARKer:SET:CENTer

Move marker to ref level  
CALCulate[:SELECTed]:MARKer:SET:REFLevel

Marker On/Off  
CALCulate[:SELECTed]:MARKer[:STATE]e

Move marker to other trace  
CALCulate[:SELECTed]:MARKer:TRACE

Move/read marker X-axis position  
CALCulate[:SELECTed]:MARKer:X

Read marker Y-axis position  
CALCulate[:SELECTed]:MARKer:Y?

Read marker Y-axis uncertainty position  
CALCulate[:SELECTed]:MARKer:Y:UNCertainty?

Markers - coupled  
CALCulate[:SELECTed]:MARKer:COUPled

Marker search - Target  
CALCulate[:SELECTed]:MARKer:FUNCTION:TARGet
Tracking On/Off  
\texttt{CALCulate[:SELected]:MARKer:FUNCtion:TRACTing}

\textbf{Calibration}

Receiver calibration (Cancel)  
\texttt{[:SENSe]:CORRection:RCALibration:CANCel}

Receiver calibration (Run)  
\texttt{[:SENSe]:CORRection:RCALibration:RUN}

Receiver calibration state (query only)  
\texttt{[:SENSe]:CORRection:RCALibration[:STATe]?}

User calibration (Cancel)  
\texttt{[:SENSe]:CORRection:UCALibration:CANCel}

User calibration interpolated state (query only)  
\texttt{[:SENSe]:CORRection:UCALibration:INTerpolated?}

User calibration (Run)  
\texttt{[:SENSe]:CORRection:UCALibration:RUN}

User calibration state (Enable/Disable)  
\texttt{[:SENSe]:CORRection:UCALibration[:STATe]}

\textbf{Read / Save Data}

Set read format  
\texttt{FORMat[DATA]}

Saves trace to CSV file.  
\texttt{MMEMory:STORe:FDATa}

Read Trace Data  
\texttt{TRACe<n>:DATA?}

Trace Data (Query the trace memory data)  
\texttt{TRACe<n>:MEMory:DATA?}

Read the uncertainty memory trace data  
\texttt{TRACe<n>:MEMory:UNCertainty:DATA?}

Read the uncertainty lower memory trace data  
\texttt{TRACe<n>:MEMory:UNCertainty:LOWer:DATA?}

Read the uncertainty upper memory trace data  
\texttt{TRACe<n>:MEMory:UNCertainty:UPPer:DATA?}

Trace Data (Query trace data)  
\texttt{TRACe<n>:UNCertainty:DATA?}

Trace Data (Query trace data lower values)  
\texttt{TRACe<n>:UNCertainty:LOWer:DATA?}

Trace Data (Query trace data upper values)  
\texttt{TRACe<n>:UNCertainty:UPPer:DATA?}

\underline{Last Modified:}  
01\textit{june}2018  
Added NF mode Opt. 356 (10.3)

\textbf{AM/FM Metrics (Option 355) Commands - Requires SA}

In this topic:
- **Metrics**

  **See Also**
  - **Commands that are Common to All Modes**
  - **Correction Methods Explained**
  - **New Programming Commands**
  - **Status Registers**

### Metrics

Set the AM/FM Metrics measurement type (Select 1 of \( n \) measurement choices)

[:SENSe]:MEASurement:ADEMod

Set the AM demodulation window's Y axis--top amplitude

[:SENSe]:ADEMod:METRics:AMTY

Select type of metrics AM FW wideband or narrow band

[:SENSe]:ADEMod:METRics:DTYPE

Set the FM demodulation window's Y axis--top amplitude

[:SENSe]:ADEMod:METRics:FMTY

Enables/disables the Audio ON or OFF while metrics enabled

[:SENSe]:ADEMod:METRics:LON

Sets the Listen time for the measurement

[:SENSe]:ADEMod:METRics:LTIMe

Enable display of Peak+ and Peak – in the demodulation window

[:SENSe]:ADEMod:METRics:MMENable

Sets the time span of the demodulation window

[:SENSe]:ADEMod:METRics:STIMe

Sets the Tune (Center) frequency

[:SENSe]:ADEMod:METRics:TFReq

Returns 8 doubles of the AM measurement (query only)

:DISPlay:ADEMod:METRics:AM:RESults:DATA?

Returns 8 doubles of the FM measurement (query only)

:DISPlay:ADEMod:METRics:FM:RESults:DATA?

---

### Commands that are Common to All Modes

In this topic:

- **Lockout Front-Panel Operation**
- **Preset / User Preset**
- **Display Control**
- **Triggering**
- **Data Transfer Format and Order**
- **Catalog and Select Mode**
• Mass Memory - Files
• External Reference Source
• Status Registers
• System
• Battery information
• GPS
• Variable Voltage Source
• Power OFF / ON
• IEEE - Common Commands

Notes:
• There is NO command to set the Security Level.
• There is NO command to set Startup Mode.
• There is NO commands to set Date/Time using Internet.

See Also
• CAT Mode Commands
• NA Mode Commands
• PM Mode Commands
• SA Mode Commands
• VVM Mode Commands
• ERTA Mode Commands
• Pulse Measurements Mode Commands
• Status Registers

Lockout Front-Panel Operation
Lockout keypresses INSTrument:GTR
Return local control INSTrument:GTL

Preset / User Preset
Preset all modes to Factory settings. SYStem:PRESet
Preset the current mode only to Factory settings. SYStem:PRESet:MODE
Preset all modes to User settings. SYStem:UPReset
Preset the current mode only to User settings. SYStem:UPReset:MODE
Save User Preset settings SYStem:UPReset:SAVE

Display Control

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display OFF</td>
<td>DISPLAY:ENABLE</td>
</tr>
</tbody>
</table>
Set brightness: `DISPLAY:BRIGHTness`
Set date format: `DISPLAY:DATE:FMT`
Change keywords: `DISPLAY:KEYWord[:DATA]`
Reset default keywords: `DISPLAY:KEYWord:DEFault`
Display marker table: `DISPLAY:TABLE:MARKer`
Set time format: `DISPLAY:TIME:FMT`
Set title string: `DISPLAY:TITLe:DATA`
Display title: `DISPLAY:TITLe:STATE`

**Triggering**
Continuous triggering: `INITiate:CONTinuous`
Single trigger: `INITiate[:IMMediate]`

**Data Transfer Format and Order**
Format: `FORMAT[:DATA]`
Byte Order: `FORMAT:BORDER`

**Catalog and Select Mode**
Read available modes: `INSTrument:CATalog?`
Set mode: `INSTrument[:SELect]`

**Mass Memory - Files**
Read files: `MMEMory:CATalog?`
Change folder: `MMEMory:CDIRectory`
Copy files: `MMEMory:COPY`
Read any file: `MMEMory:DATA`
Delete file: `MMEMory:DElete`
Recall an instrument state file: `MMEMory:LOAD:STATe`
Make a new folder: `MMEMory:MDIRectory`
Rename a file: `MMEMory:MOVE`
Remove a folder: `MMEMory:RDIREctory`
Save a picture file: `MMEMory:STORe:IMAGe`
Save an instrument state file: `MMEMory:STORe:STATe`
See Also (Mode-specific MMEM commands): `MMEMory:STORe:FDATA`
Save SNP data: MMMory:STORe:SNP[:DATA]
Recall antenna data: MMMory:LOAD:ANTenna
Recall cable data: MMMory:LOAD:CABLE
Save antenna data: MMMory:STORe:ANTenna
Save cable data: MMMory:STORe:CABLE

**External Reference Source**

Set external reference source: [:SENSe]:ROSCillator:SOURce
Read external source status: [:SENSe]:ROSCillator:STATus?

**Status Registers**

Read RecordPlayback status: STATus:OPERation:SAMode:CONDition?
Read external source status: STATus:QUEStionable:FREQuency:CONDition?
Read ADC Overrange errors: STATus:QUEStionable:INTegrity:CONDition?
Read limit line failure: STATus:QUEStionable:LIMit:CONDition?

**Battery Information**

Read presence of battery: SYSTem:BATTery
Read absolute charge: SYSTem:BATTery:ABSCharge?
Read average current flow: SYSTem:BATTery:ACURrent?
Read remaining run time: SYSTem:BATTery:ARTTe?
Read chemistry type: SYSTem:BATTery:CHEMistry?
Read current flow: SYSTem:BATTery:CURRent?
Read number of charge cycles the battery has experienced: SYSTem:BATTery:CYCLes?
Read manufacture date of the battery: SYSTem:BATTery:DATE?
Read capacity: SYSTem:BATTery:FCAPacity?
Read accuracy of the battery gauge: SYSTem:BATTery:MAXError?
Read manufacturer name: SYSTem:BATTery:MFGname?
Read remaining battery capacity: SYSTem:BATTery:RCAPacity?
Read current charge compared to full capacity: SYSTem:BATTery:RELCharge?
Read remaining run time: SYSTem:BATTery:RTTE?
Set and read battery saver state. OFF leaves the source ON between sweeps.  

```
SYSTem:BATTery:SAVer
```

Read serial number of the battery.  

```
SYSTem:BATTery:SN?
```

Read use status  

```
SYSTem:BATTery:STATus
```

Read battery temperature  

```
SYSTem:BATTery:TEMPerature?
```

Read vendor / distributor of the battery.  

```
SYSTem:BATTery:VENDor?
```

Read battery voltage.  

```
SYSTem:BATTery:VOLTage?
```

**System**

Set and read the system date  

```
SYSTem:DATE
```

Set and read the system time  

```
SYSTem:TIME
```

Immediately erase all user data  

```
SYSTem:ERASe
```

Immediately erase the error log  

```
SYSTem:ERRor:LOG:ERASe
```

Read the FieldFox error queue  

```
SYSTem:ERRor[:NEXT]?
```

Reset default preferences  

```
SYSTem:PREFerences:DFLT
```

Save system preferences  

```
SYSTem:PREFerences:SAVE
```

Set time zone.  

```
SYSTem:TZONe
```

Catalog time zones.  

```
SYSTem:TZONe:CATalog?
```

Set and read system volume  

```
SYSTem:AUDio:VOLume
```

Set and read system volume mute state  

```
SYSTem:AUDio:MUTe
```

Read the SCPI version  

```
SYSTem:VERSion?
```

**GPS**

Set and read GPS ON|OFF state.  

```
SYSTem:GPS[:STATe]
```

Query returns a string containing Carrier to Noise (C/No dBHz) data.  

```
SYSTem:GPS:CNOise?
```

Read the locked state.  

```
SYSTem:GPS:LSTate?
```

Set and read the display state.  

```
SYSTem:GPS:DISPlay:STATe
```

Set and read the clock sync state.  

```
SYSTem:GPS:SYNChronize
```

Set and read the lat/long format.  

```
SYSTem:GPS:DISPlay:COORDinate:FORMat
```
Set and read the distance units: \texttt{SYSTem:GPS:DISPlay:DISTance:UNIT}

Read the last locked reading: \texttt{SYSTem:GPS:DATA:LAST?}

Read the current GPS data: \texttt{SYSTem:GPS:DATA?}

**Variable Voltage Source**

Current draw: \texttt{SYSTem:VVS:CURRent?}

Enable Voltage Source: \texttt{SYSTem:VVS:ENABLE}

Set the high impedance load state: \texttt{SYSTem:VVS:HIMD}

Max current draw: \texttt{SYSTem:VVS:MAXCurrent?}

Read measured voltage: \texttt{SYSTem:VVS:MVOLtage?}

Read state (On/ Off/ Tripped): \texttt{SYSTem:VVS:[STATe]?}

Read the present requested voltage: \texttt{SYSTem:VVS:RVOLtage?}

Set voltage: \texttt{SYSTem:VVS:VOLTage}

**Power OFF / ON**

Automatically turns the FieldFox ON when power is applied: \texttt{SYSTem:PWR:AUTO}

Turns the FieldFox OFF: \texttt{SYSTem:PWR:SHUTdown}

Sets delay before turning the FieldFox OFF: \texttt{SYSTem:PWR:SHUTdown:DLY}

Set time to wait before rebooting the FieldFox: \texttt{SYSTem:PWR:SHUTdown:DURation}

Reads whether the DC supply is connected: \texttt{SYSTem:DCSupply?}

**IEEE - Common Commands**

Clear status: \texttt{*CLS}

Event Status Enable: \texttt{*ESE}

Event Status Enable Query: \texttt{*ESR?}

Identify: \texttt{*IDN?}

Operation complete command: \texttt{*OPC}

Operation complete query: \texttt{*OPC?}

Identify Options Query: \texttt{*OPT?}

Reset: \texttt{*RST}

Wait: \texttt{*WAI}
Calibration Commands

See Also
- Correction Methods Explained
- Calibration Examples
- Status Registers

Set and read error term data  [:SENSe]:CORRection:COEFFicient[:DATA]
Read number of cal steps  [:SENSe]:CORRection:COLLect:GUIDed:SCOunt
Measure step number  [:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire
Prompt for step number  [:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMpt
Measure Quick Cal  [:SENSe]:CORRection:COLLect[:ACQuire]:INT
Measure load  [:SENSe]:CORRection:COLLect[:ACQuire]:LOAD
Measure open  [:SENSe]:CORRection:COLLect[:ACQuire]:SHORt
Measure thru  [:SENSe]:CORRection:COLLect[:ACQuire]:THRU
Set Cal Kit  [:SENSe]:CORRection:COLLect:CKIT:LABel
Catalog all cal kits  [:SENSe]:CORRection:COLLect:CKIT:LABel:CATalog?
Set connectors  [:SENSe]:CORRection:COLLect:CONNECTor
Select method - Enhanced Response  [:SENSe]:CORRection:COLLect:METHod:ERES
Select method - QuickCal  [:SENSe]:CORRection:COLLect:METHod:QCALibrate:CALibrate
Select method - QuickCal Enhanced Response  [:SENSe]:CORRection:COLLect:METHod:QCALibrate:ERESponse
Select method - Simple Open response  [:SENSe]:CORRection:COLLect:METHod[:RESPonse]:OPEN
Select method - Simple Short response  [:SENSe]:CORRection:COLLect:METHod[:RESPonse]:SHORt
Select method - Thru response  [:SENSe]:CORRection:COLLect:METHod[:RESPonse]:THRU
Select method - Short response
[:SENSe]:CORRection:COLLect:METHod:SRESponse

Select method - Open response
[:SENSe]:CORRection:COLLect:METHod:ORESponse

Select method - 1-port SOLT
[:SENSe]:CORRection:COLLect:METHod:SOLT1

Select method - TRL
[:SENSe]:CORRection:COLLect:METHod:TRL

Read method
[:SENSe]:CORRection:COLLect:METHod:TYPE?

Set AutoOrient for ECal
[:SENSe]:CORRection:COLLect:ECAL:AORient

Set simple ECal
[:SENSe]:CORRection:COLLect:ECAL:SIMPle

Omit Isolation
[:SENSe]:CORRection:COLLect:OISolation

Select Medium
[:SENSe]:CORRection:MEDium

Set Waveguide cutoff
[:SENSe]:CORRection:WGCutoff

Select Waveguide standard (CAT only)
[:SENSe]:CORRection:WAVeguide:STANdard

Finish Cal
[:SENSe]:CORRection:COLLect:SAVE

Turn ALL Correction ON and OFF
[:SENSe]:CORRection[:STATe]

Turn User Correction ON and OFF
[:SENSe]:CORRection:USER[:STATe]

Set system impedance
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]

Set CalReady type
[:SENSe]:CORRection:CALReady:TYPE

Last Modified:
11sep2017  Added this topic

Status Registers

This topic contains images of the FieldFox status registers.

This topic contains the following registers:
- Summary Status
- Status Questionable Alignment
- Status Questionable Calibration
- Status Questionable Frequency
- Status Questionable Integrity
- Status Questionable Limit
- Status Questionable
- Status Operation SA Mode
- Status Operation

Field Fox Status Registers Summary

- Status Byte Register
- Unused
- Unused
- Error/Event Queue Summary
- Questionable Summary
- Message Available (MAV)
- Standard Event Summary
- Req Serv Summary (RDS)
- Operation Summary

- Service Request Enable Register

- Condition Register
- Positive Transition Register
- Negative Transition Register
- Event Enable Register
- Event Register

- STATus:QUESTionable:ALIGNment

- SA InstaAlign Needed
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

- SA IF Flatness Align Needed
Last Modified:

22oct2017  Added this topic
New Programming Commands

The following is a list of new commands for each major release:

**A.10.3x**
See NF Mode Commands NEW!
See also programming Python programming example and MATLAB examples (VISA and LAN)
In the Common Commands topic:
SYS TEM:VVS:HIMD
SYS TEM:VVS:RVOLTage?

**A.10.1x**
See IQA Mode Commands
See also Status Registers
See also Memory Command Examples

**A.09.53**
CALCulate:MARKer:STRack

**A.09.50**
[:SENSe]:ACQuisition:TIME
[:SENSe]:ACQuisition:TIME:AUTO
[:SENSe]:DENSity:BPLevel
[:SENSe]:DENSity:RPLevel
[:SENSe]:DETe ctor:TRACE(1:4):FUNCTION
[:SENSe]:FREQuency:SPAN:Bandwidth:RESolution:RATio?
[:SENSe]:IFFLatness:ALIGNment:STATE
[:SENSe]:MEASure
[:SENSe]:SPECTrogram:TPDivision
:DISPlay:VIEW:DENSity:PERSistence
:DISPlay:VIEW:DENSity:STATE
:TRACe:PRESet:ALL
:TRIGger[:SEQUence]:FRAMe:OFFSet
:TRIGger[:SEQUence]:FRAMe:OFFSet:DISPlay:RESet
:TRIGger[:SEQUence]:FRAMe:PERiod
:TRIGger[:SEQUence]:HOLDoff
:TRIGger[:SEQUence]:LEVEL
See also, RTSA Mode Commands

**A.09.25**
A.08.15 and A.09.15

AM/FM Metrics

[:SENSe]:MEASurement:AEMod
[:SENSe]:AMEMod:METRics:AMTY
[:SENSe]:AMEMod:METRics:DTYPe
[:SENSe]:AMEMod:METRics:FMTY
[:SENSe]:AMEMod:METRics:LON
[:SENSe]:AMEMod:METRics:LTIMe
[:SENSe]:AMEMod:METRics:MMENable
[:SENSe]:AMEMod:METRics:STIMe
[:SENSe]:AMEMod:METRics:TFReq

:DISPlay:AMEMod:METRics:AM:RESults:DATA?
:DISPlay:AMEMod:METRics:FM:RESults:DATA?

Channel Scanner

:CHSCan:DATA?
:CHSCan:DISPlay:SORT
:CHSCan:DISPlay:SORT:ORDer
:CHSCan:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision
:CHSCan:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel

:CHSCan:EDIT:LIST?
:CHSCan:EDIT:LIST:ADD
:CHSCan:EDIT:LIST:CLEar
:CHSCan:EDIT:RANGe:COUNt
:CHSCan:EDIT:RANGe:IBW
:CHSCan:EDIT:RANGe:SPAN
:CHSCan:EDIT:RANGe:STARt
:CHSCan:EDIT:RANGe:STEP

:CHSCan:LOG:ACTion:RECord
:CHSCan:LOG:ACTion:STOP
:CHSCan:LOG:CONFigure:INTerval:DISTance
:CHSCan:LOG:CONFigure:INTerval:STATe
:CHSCan:LOG:CONFigure:INTerval:TIMe
:CHSCan:LOG:CONFigure:INTerval:TYPe
:CHSCan:LOG:CONFigure:SAVe:AUTo[:STATe]
:CHSCan:LOG[:STATe]
A.08.04
ERTA Mode
[:SENSe]:MEASurement:ERTA:PNID
[:SENSe]:MEASurement:ERTA:PVERify?

GPS Mode
SYSTem:GPS:CNOise?

A.08.00
All Modes
SYSTem:PREFerences:SNP

CAT Mode
CALCulate[:SELeCted]:SMOothing:APERture
CALCulate[:SELeCted]:SMOothing[:STATe]

CAT - TDR
CALCulate:PARameter:DEFine
CALCulate[:SELeCted]:MARKer:TDR:FORMat
[:SENSe]:SWEep:TDR:AUTO:FREQuency:STOP:MAXimum
[:SENSe]:SWEep:TDR:RES

NA Mode
CALCulate[:SELeCted]:OFFSet:SLOPe
CALCulate[:SELeCted]:OFFSet[:MAGNitude]

ERTA Mode
[:SENSe]:MEASurement:ERTA:PIP
[:SENSe]:MEASurement:ERTA:VPIP?
[:SENSe]:MEASurement:ERTA:ROLE?
[:SENSe]:MEASurement:ERTA:PSTatus
TRACe:ERTA:RINPut

SA Mode
Source Tracking Offset
SOURce:TOFFset:ENABle
SOURce:TOFFset:FREQuency
SOURce:TOFFset:REVerse

Reverse sweep (Opt. 209 only)
[:SENSe]:SWEep:FREVerse

Select freq axis annotation
[:SENSe]:FREQuency:ANNotation[:SELeCt]

Read ADC over-range status
CALCulate:MEASurement:WAOR?

Source-side corrections
MMEMory:LOAD:SANTenna
MMEMory:LOAD:SCABle
MMEMory:STORe:SANTenna
MMEMory:STORe:SCABle
[:SENSe]:AMPLitude:CORRections:SANTenna:DEFault
[:SENSe]:AMPLitude:CORRections:SANTenna[:STATe]
[:SENSe]:AMPLitude:CORRections:SCABle:DEFault
[:SENSe]:AMPLitude:CORRections:SCABle[:STATe]

A.07.75
All Modes
INSTrument:GTL
INSTrument:GTR

NA Mode
Big Marker Display States (A and B)

VVM Mode
[:SENSe]:CORRection:ZERO:REFerence

A.07.50
Frequency Offfset Power Sensor (FOPS) commands
Built-in Power Meter (CPM) Mode Commands
MMEMory:RDIRectory - Added optional argument

SA Mode
[:SENSe]:SWEep:MTIMe?
Renamed commands:

- SOURce:ENABle
- SOURce:FREQuency[CW]
- SOURce:MODE
- SOURce:NORMalize
- SOURce:POWer
- SOURce:POWer:MAXimum

**NA Mode**

- CALCulate:PARameter:DEFine (New arguments)
- CALCulate[SELECTed]:CONVersion:FUNCtion

**CAT Mode**

- [:SENSe]:CORRection:WAVeguide:STANdard

**CAT and NA Mode**

- [:SENSe]:CORRection:COLLect:METHod:SRESponse
- [:SENSe]:CORRection:COLLect:METHod:ORESponse

---

### A.07.25

**Pulse Measurement Mode Commands**

**CAT and NA modes**

- [:SENSe]:CORRection:COLLect:CKIT:LABel <ECal module>
- [:SENSe]:CORRection:COLLect:ECAL:AORient
- [:SENSe]:CORRection:COLLect:ECAL:SIMPle
- [:SENSe]:CORRection:COLLect:METHod:TRL
- [:SENSe]:CORRection:COLLect:OISolation
- [:SENSe]:CORRection:MEDium
- [:SENSe]:CORRection:WGCutoff

**SA Mode**

- CALCulate:MARKer:TZERo:FIXed

**USB Power Meter Mode**

- SOURce:ENABle
- SOURce:POWer

**Common to ALL Modes**

- FORMat:BORDer

**User Preset**

- SYSTem:UPReset
- SYSTem:UPReset:FPANel[:STATe]
- SYSTem:UPReset:MODE
- SYSTem:UPReset:SAVE
A.07.00

**NA mode**
- TRIGger:SOURce
- TRIGger:SLOPe
- CALCulate[:SElected]:MARKer:FORMat

**SA mode**
- TRIGger[:SEQuence]:SOURce *(New argument)*
- CALCulate:MEASurement:QAMplitude
- [:SENSe]:BURSt:ALIGNment:NOW
- [:SENSe]:BURSt:ALIGNment[:STATe]
- TRIGger[:SEQuence]:POSition
- TRIGger[:SEQuence]:POSition:STATe
- TRIGger[:SEQuence]:DELay:STATe
- TRIGger[:SEQuence]:FGATe:DELay
- TRIGger[:SEQuence]:FGATe:VIEW:TIME
- TRIGger[:SEQuence]:FGATe:VIEW[:STATe]
- TRIGger[:SEQuence]:FGATe:WIDTH
- TRIGger[:SEQuence]:FGATe[:STATe]

**Multiple modes**
- CALCulate[:SElected]:LIMit:SOUNd

A.06.17

**Multiple modes**
- SYSTem:PWR:AUTO
- SYSTem:DCSupply?

**CAT / NA mode**
- [:SENSe]:CORRection:CALReady:TYPE
- [:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire
- [:SENSe]:CORRection:COLLect:GUIDed:SCOunt
- [:SENSe]:CORRection:COLLect:GUIDed:STEP:PROMpt

A.06.00

**Multiple modes**
- CALCulate:LIMit:LLData
- STATus:QUEStionable:LIMit:CONDition?
- SYSTem:VVS:CURRent?
- SYSTem:VVS:ENABle
- SYSTem:VVS:MAXCurrent?
SYSTem:VVS:MVOLTage?
SYSTem:VVS[STATE]?
SYSTem:VVS:VOLTage
SYSTem:DCSupply?

NA mode
[:SENSe]:AVERage:MODE

Time Domain Transform
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:CENTer
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:SHApe
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:SPAN
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:STARt
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:STATE
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:STOP
CALCulate[:SELECTed]:FILTER[:GATE]:TIME:TYPE
CALCulate[:SELECTed]:TRANSform:TIME: [TYPE]
CALCulate[:SELECTed]:TRANSform:TIME:CENTer
CALCulate[:SELECTed]:TRANSform:TIME:IMPulse:WIDTH
CALCulate[:SELECTed]:TRANSform:TIME:KBEssel
CALCulate[:SELECTed]:TRANSform:TIME:LPFRequency
CALCulate[:SELECTed]:TRANSform:TIME:SPAN
CALCulate[:SELECTed]:TRANSform:TIME:STARt
CALCulate[:SELECTed]:TRANSform:TIME:STATE
CALCulate[:SELECTed]:TRANSform:TIME:STEP:RTIMe
CALCulate[:SELECTed]:TRANSform:TIME:STIMulus
CALCulate[:SELECTed]:TRANSform:TIME:STOP

CAT mode
CALCulate:PARameter:SELECT
DISPlay:WINDow:SPLit
CALCulate:MARKer:COUPle

SA mode
[:SENSe]:AMPLitude:CORRections:ANTenna:DEFault
[:SENSe]:AMPLitude:CORRections:CABLE:DEFault
[:SENSe]:AMPLitude:ALIGNment[:STATE]
[:SENSe]:AMPLitude:ALIGNment:NOW
[:SENSe]:BANDwidth:IF:OUT

Power Meter Mode
[:SENSe]:AVERage:SDETect

A.05.50
DISPlay:WINDow:TRACe:Y:DLINe <level>
DISPlay:WINDow:TRACe:Y:DLINe:STATe <ON/OFF>
CALCulate:MARKer:BWIDth:DATA
CALCulate:MARKer:BWIDth[:STATe]
CALCulate:MARKer:FUNCtion:BWIDTH:THreshold
CALCulate:MARKer:FUNCtion:TRACking
CALCulate:MARKer:FUNCtion:TARGet
SYSTem:AUDio:MUTe
SYSTem:AUDio:VOLume
[:SENSe]:RADio:STANdard[:SELect]
[:SENSe]:AVERage:CLEAR

A.05.30
CALCulate:MARKer:FUNCtion
CALCulate:MARKer:FUNCtion:BAND:SPAN
CALCulate:MARKer:FUNCtion:BAND:SPAN:AUTO
CALCulate:MARKer:SET
CALCulate[:SELECTed]:SMOothing:APERture
CALCulate[:SELECTed]:SMOothing[:STATe]
DISPlay[:WINDow][:NUMeric]:RESolution
[:SENSe]:BWID
[:SENSe]:CORRection:EXTension:PORT1
[:SENSe]:CORRection:EXTension:PORT2
[:SENSe]:CORRection:EXTension[:STATe]
[:SENSe]:ISOurce:ENABle
[:SENSe]:ISOurce:FREQuency[:CW]
[:SENSe]:ISOurce:MODE
[:SENSe]:ISOurce:POWer
SYSTem:TZONe
SYSTem:TZONe:CATalog?

Replacement Commands

The following is a list of commands that have been replaced.

Superseded Commands
Superseded commands will continue to work in existing programs. However, the replacement command usually has more functionality and is recommended.

<table>
<thead>
<tr>
<th>Superseded command</th>
<th>Replacement command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCulate:MARKer:NOISe[:STATe]</td>
<td>CALCulate:MARKer:FUNCtion</td>
</tr>
</tbody>
</table>
#### OBSOLETE commands
These commands will NOT continue to work in existing programs.

<table>
<thead>
<tr>
<th>Obsolete command</th>
<th>Replacement command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SENSe]:TAListen:AVOLume (A.05.50)</td>
<td>SYSTem:AUDio:VOLume</td>
</tr>
<tr>
<td>[:SENSe]:RADio:STANdard (A.05.50)</td>
<td>[:SENSe]:RADio:STANdard[:SELeCt]</td>
</tr>
</tbody>
</table>

---

**Common Commands**
The following IEEE 488.2 Common Commands can be used with the FieldFox:

- **CLS**
  Clears the instrument status byte by emptying the error queue and clearing all event registers. Also cancels any preceding *OPC command or query.

- **ESE - Event Status Enable**
  Sets bits in the standard event status enable register.

- **ESE? - Event Status Enable Query**
  Returns the results of the standard event enable register. The register is cleared after reading it.

- **ESR? - Event Status Enable Register**
  Reads and clears event status enable register.

- **IDN? - Identify**
  Returns a string that uniquely identifies the FieldFox. The string is of the form "Keysight Technologies", <model number>, <serial number>, <software revision> and so forth.

- **OPC - Operation complete command**
Generates the OPC message in the standard event status register when all pending overlapped operations have been completed (for example, a sweep, or a Default).

*OPC? - Operation complete query

Returns an ASCII "+1" when all pending overlapped operations have been completed.

*OPT? - Identify Options Query

Returns a string identifying the analyzer option configuration.

*RST - Reset

Executes a device reset and cancels any pending *OPC command or query. All trigger features in HOLD. The contents of the FieldFox non-volatile memory are not affected by this command.

*SRE - Service Request Enable

Before reading a status register, bits must be enabled. This command enables bits in the service request register. The current setting is saved in non-volatile memory.

*SRE? - Service Request Enable Query

Reads the current state of the service request enable register. The register is cleared after reading it.

*WAI - Wait

Prohibits the instrument from executing any new commands until all pending overlapped commands have been completed.

CALCulate:FEED:MODE <char>

(Read-Write) Set and query the current measurement.

Relevant Modes

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Pulse Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;char&gt;</td>
<td>Measurement parameter. Choose from:</td>
</tr>
<tr>
<td></td>
<td>PEAK - Peak (Meter-style)</td>
</tr>
<tr>
<td></td>
<td>AVER - Average (Meter-style)</td>
</tr>
<tr>
<td></td>
<td>PTAV - Peak to Average (Meter-style)</td>
</tr>
<tr>
<td></td>
<td>TMOD - Trace Graph mode</td>
</tr>
</tbody>
</table>

Examples

CALC:FEED:MODE AVER

Query Syntax

CALCulate:FEED:MODE?

Return Type

Character

Default

PEAK

Last Modified:

30-Oct-2013 New command

CALCulate:IREJection:LEVel <char>

(Read-Write) Set and query the Interference rejection level.
Relevant Modes CAT

Parameters
<char> Interference rejection level. Choose from:
  OFF – no interference rejection.
  LEV1 – level 1
  LEV2 – level 2
  LEV3 – level 3

Examples CALC:IREJ:LEV LEVEL

Query Syntax CALCulate:IRejction:LEVel?

Return Type Character

Default OFF

Last Modified:
1-Aug-2011 New command

CALCulate:MARKer:AUDio:BEEP <bool>

(Read-Write) Set or return the state of an Audio Beep on the active SA mode marker. Audio Beep ON creates a marker if not already ON. Set beep volume using SYSTem:AUDio:VOLUME <num>.

Relevant Modes SA

Parameters
<bool> Choose from:
  ON (or 1) - Audio beep ON
  OFF (or 0) - Audio beep OFF

Examples CALC:MARK:AUD:BEEP ON

Query Syntax CALCulate:MARKer:AUDio:BEEP?

Return Type Boolean

Default Off

Last Modified:
1-Aug-2011 New command (A.05.50)

CALCulate:MARKer<n>:DREF:FIXed <bool>

(Read-Write) Set and query the state of fixed delta / reference markers. Created delta markers using CALCulate:MARKer[:STATe]
Relevant Modes  SA, RTSA

Parameters

<n> Existing marker to fix or let float. Choose from 1 to 6.

<bool> Choose from:

OFF or 0 - Reference marker floats with each sweep at the Y-axis position of the data trace

ON or 1 - Reference marker is fixed at the Y-axis position of the data trace when the marker was created.

Examples  CALC:MARK2:DREF:FIX OFF

Query Syntax  CALCulate:MARKer<n>:DREF:FIXed?

Return Type  Boolean

Default  ON

Last Modified:

10-june-2016  Added RTSA mode (9.50)

CALCulate:MARKer<n>:FCOunt[:STATe] <bool>

(Read-Write) Set and query the Frequency counter marker ON/OFF state.

Use CALCulate:MARKer[:STATe] to first create a marker.

Use CALCulate:MARKer:X to move the marker to the frequency of interest.

Use CALCulate:MARKer:FCOunt:X? to read the frequency counter marker.

Relevant Modes  SA

Parameters

<n> Marker number to become a frequency counter marker. A marker is created if not already ON. Choose from 1 to 6. If unspecified, value is set to 1.

<bool> Frequency counter marker state. Choose from:

ON (or 1) - Frequency counter marker ON.

OFF (or 0) - Frequency counter marker OFF.

Examples  CALC:MARK2:FCO 1

Query Syntax  CALCulate:MARKer<n>:FCOunt?

Return Type  Boolean

Default  OFF

Last Modified:

20-Oct-2010  New command (5.30)
CALCulate:MARKer:FCOunt:X?

(Read-only) Read the frequency of the frequency counter marker in Hz. Use CALCulate:MARKer:FCOunt[:STATe] to make a marker a frequency counter marker.

Caution: For firmware A.09.59 and greater the returned x-axis value will now return up to 15 decimal places to the right of the decimal for distance x-axis or time x-axis units, instead of truncating all digits to the right of the decimal.

Example 1: Old DTF x-axis format for 82.5 meters would return 83. New x-axis format returns: 8.250000000E+1.

Example 2: Old RL measured marker x-axis value for 2.1862505 GHz would return 2186250500. New marker x-axis value returns: 2.186250500E+10.

Be aware that as a result some software content may need to be modified to accept this new behavior.

Relevant Modes SA
Parameters None
Examples CALC:MARK:FCO:X?
Return Type Numeric
Default N/A

Last Modified:
20-Oct-2010 New command (5.30)

CALCulate:MARKer<n>:FUNCtion <char>

(Read-Write) Causes the specified marker to become one of the SA Marker functions. Also causes the specified marker to be turned ON if it is not already.

Relevant Modes SA
Parameters <n> Existing marker to become a marker function. Choose from 1 to 6.
<char> Marker function. Choose from:
- OFF - Marker is returned to it's previous state (normal or delta).
- NOISe - Marker becomes a noise marker.
- BPOWER - Marker becomes a Band/Interval Power marker.

For non-zero span measurements, a Band Power marker integrates total power over the Band Power Span, which is set using:
- CALCulate:MARKer:FUNCtion:BAND:SPAN and
- CALCulate:MARKer:FUNCtion:BAND:SPAN:AUTO.

For Zero-span measurements, an Interval Power marker calculates the average power over the Interval Power Span, which is set using:
- CALCulate:MARKer:FUNCtion:BAND:SPAN and
- CALCulate:MARKer:FUNCtion:BAND:SPAN:AUTO.

Examples CALC:MARK1:FUNC NOIS
**CALCulate:MARKer:FUNCTION:BAND:SPAN** `<num>`

(Read-Write) Set and read the frequency span for ALL SA mode Band Power Markers.
Set CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO to OFF.

**Relevant Modes** SA

**Parameters**

<`num>` Band power markers frequency span in Hz. Choose a value equal to or less than the FieldFox frequency span.

**Examples**

CALC:MARK:FUNC:BAND:SPAN 1e6 'Set span to 1 MHz

**Query Syntax** CALCulate:MARKer:FUNCTION:BAND:SPAN?

**Default** 5% of existing frequency span.

---

**CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO** `<bool>`

(Read-Write) Set and read the method by which the frequency span for ALL SA mode Band Power Markers is set.

**Relevant Modes** SA

**Parameters**

<`bool>` Band power frequency span method.

ON (or 1) - Band Power frequency span is always 5% of existing frequency span.

OFF (or 0) - Set Band Power frequency span using CALCulate:MARKer:FUNCTION:BAND:SPAN

**Examples** CALC:MARK:FUNC:BAND:SPAN:AUTO 1

**Query Syntax** CALCulate:MARKer<n>:FUNCTION:BAND:SPAN:AUTO?

**Default** ON
CALCulate:MARKer:FUNCtion:INTerval:SPAN <num>

(Read-Write) Set and read the time interval for ALL SA mode Interval Markers. Set CALCulate:MARKer:FUNCtion:INTerval:SPAN:AUTO to OFF.

Relevant Modes  SA

Parameters

<num> Interval span in seconds. Choose a value between 1e-9 and 100 seconds.

Examples  CALC:MARK:FUNC:INT:SPAN 1e-3 'Set span to 1 mSec

Query Syntax  CALCulate:MARKer:FUNCtion:INTerval:SPAN?

Default  5% of existing X-axis span.

CALCulate:MARKer:FUNCtion:INTerval:SPAN:AUTO <bool>

(Read-Write) Set and read the method by which the time span for ALL SA mode Interval Markers is set.

Relevant Modes  SA

Parameters

<bool> Interval span method.

ON (or 1) - Interval time span is always 5% of X-axis.
OFF (or 0) - Set Interval time span using CALCulate:MARKer:FUNCtion:INTerval:SPAN

Examples  CALC:MARK:FUNC:INT:SPAN:AUTO 1

Query Syntax  CALCulate:MARKer<n>FUNCtion:INTerval:SPAN:AUTO?

Default  ON
### CALCulate[:SELected]:MARKer<n>:FUNCtion:PLEFt

(Write-Only) Causes the specified marker to find the next data point to the left that meets the 'Peak' criteria. When no data points to the left meet the Peak criteria, the marker does not move.

**Relevant Modes**
- NF, SA, RTSA

**Note:** SA & RTSA modes do NOT recognize the optional [:SELected] node.

**Parameters**
- `<n>` New or existing marker to move. Choose from 1 to 6.

**Examples**
- `CALC:MARK1:FUNC:PLEF`

**Query Syntax**
- Not Applicable

**Default**
- Not Applicable

---

**Last Modified:**
- 01june2018  Added NF mode Opt. 356 (10.3)
- 10-june-2016  Added RTSA mode (9.50)

### CALCulate[:SELected]:MARKer<n>:FUNCtion:PRIGht

(Write-Only) Causes the specified marker to search to the right of the current location for the next data point that meets the 'Peak' criteria.

**Relevant Modes**
- NF, SA, RTSA

**Parameters**
- `<n>` New or existing marker to move. Choose from 1 to 6.

**Examples**
- `CALC:MARK1:FUNC:PRIG`

**Query Syntax**
- Not Applicable

**Default**
- Not Applicable

---

**Last Modified:**
- 01june2018  Added NF mode Opt. 356 (10.3)
- 10-june-2016  Added RTSA mode (9.50)

### CALCulate:MARKer<n>:NOISe[:STATe] <bool> - Superseded

**Note:** This command is replaced with `CALCulate:MARKer:FUNCtion`.

(Read-Write) Set and query the ON|OFF state of the SA noise marker. A marker must first be created. This command then converts it to a Noise marker.

**Relevant Modes**
- SA

**Parameters**
<n> Existing marker to make a noise marker. Choose from 1 to 6.

<bool> Choose from:

- OFF - Noise marker OFF
- ON - Noise marker ON

**Examples**

```
CALC:MARK:NOIS ON
```

**Query Syntax**

```
CALCulate:MARKer:NOISe:STATe?
```

**Return Type**

Boolean

**Default**

OFF

---

**CALCulate[:SELected]:MARKer<n>:SET:CENTer**

*(Write-Only)* The center frequency of the measurement becomes the value of the specified marker. The frequency span is adjusted accordingly.

**CALCulate[:SELected]:MARKer:SET** performs the same operation.

**Relevant Modes**

NF, SA, RTSA

**Parameters**

<n> Existing marker from which the center frequency will be set. Choose from 1 to 6.

**Examples**

```
CALC:MARK1:SET:CENT
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

**CALCulate[:SELected]:MARKer<n>:SET:REFLevel**

*(Write-Only)* The reference level of the current window becomes the value of the specified marker.

**CALCulate[:SELected]:MARKer:SET** performs the same operation.

**Relevant Modes**

NF, SA, RTSA

**Parameters**

<n> Existing marker for which reference level will be set. Choose from 1 to 6.

**Examples**

```
CALC:MARK1:SET:REFL
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

Last Modified:

- 01june2018 Added NF mode Opt. 356 (10.3)
- 10-june-2016 Added RTSA mode (9.50)
CALCulate:MARKer:STRack

(Write-Read) Enables signal tracking utilizing the specified marker {1-6, default is 1}. If the specified marker is not already on, it is activated as a Normal marker. This marker then tracks the peak signal, via automatic changes to the Center Frequency. Any other markers that are active maintain their frequency position unless limited by Start or Stop Frequency.

Note: Signal tracking is incompatible (inactive) with Zero Span.

**Relevant Modes** SA

**Parameters**

<bool>  
ON (1) - Enables signal tracking utilizing the specified marker, activating it if necessary.  
OFF (2) - Disables signal tracking.

**Examples**  
CALC:MARK2:STR 1 'Marker 2 is set as the active signal tracking marker.'

**Query Syntax**  
CALC:MARK:STR?

**Default**  
0

---

CALCulate:MARKer:TZERo:FIXed <bool>

(Read-Write) Set and query the ON|OFF state of the Time Zero Fixed setting.

This feature was created to allow recall of vintage instrument states (older than Rev. 7.0) that included Zero span sweep with a trigger delay and at least one marker. Before Rev. 7.0, these instrument states were saved and recalled with the equivalent of the ON state of this setting.

**Relevant Modes** SA

**Parameters**

<bool> Choose from:  
OFF - Time zero fixed setting OFF  
ON - Time zero fixed setting ON

**Examples**  
CALC:MARK:TZER:FIX ON
**Query Syntax**

CALCulate:MARKer:TZERo:FIXed?

**Return Type**

Boolean

**Default**

OFF

---

Last Modified:

2-Apr-2014  New command

---

**CALCulate:MEASure:DATA?**

*(Read-Only)* Reads data from the current channel measurement (Channel Power, Occupied Bandwidth, or Adjacent Channel Power).

- The number of values that are returned depends on the type of channel measurement.
- The units for the values depend on the currently displayed units.
- Offsets that are not defined return invalid data.

Data is returned in the following format:

- Main channel - main channel power in dBm.
- Main channel - main channel Power Spectral Density (PSD) in dBm/Hz.
- Main channel - relative power to main channel power (this value is always zero).

- Lower Offset Frequency (1) - channel power for lower offset 1 in dBm.
- Lower Offset Frequency (1) - PSD for lower offset 1 in dBm/Hz.
- Lower Offset Frequency (1) - relative power of lower ACPR for offset 1 in dBc or dB.

- Upper Offset Frequency (1) - channel power for upper offset 1 in dBm.
- Upper Offset Frequency (1) - PSD for upper offset 1 in dBm/Hz.
- Upper Offset Frequency (1) - relative power of upper ACPR for offset 1 in dBc or dB.

- Lower Offset Frequency (2) - channel power for lower offset 2 in dBm.
- Lower Offset Frequency (2) - PSD for lower offset 2 in dBm/Hz.
- Lower Offset Frequency (2) - relative power of lower ACPR for offset 2 in dBc or dB.

- Upper Offset Frequency (2) - channel power for upper offset 2 in dBm.
- Upper Offset Frequency (2) - PSD for upper offset 2 in dBm/Hz.
- Upper Offset Frequency (2) - relative power of upper ACPR for offset 2 in dBc or dB.

- Lower Offset Frequency (3) - channel power for lower offset 3 in dBm.
- Lower Offset Frequency (3) - PSD for lower offset 3 in dBm/Hz.
- Lower Offset Frequency (3) - relative power of lower ACPR for offset 3 in dBc or dB.

- Upper Offset Frequency (3) - channel power for upper offset 3 in dBm.
- Upper Offset Frequency (3) - PSD for upper offset 3 in dBm/Hz.
- Upper Offset Frequency (3) - relative power of upper ACPR for offset 3 in dBc or dB.

**Relevant Modes**

SA
**Examples**

The following data is returned for an ACPR channel measurement. Offsets that are not defined return invalid data.

`CALC:MEAS:DATA?`

* With only one defined offset, returns
  -6.73047890E+01,-1.303150890E+02,0.0000000E+00,
  -6.78255545E+01,-1.308358553E+02,-5.207664E-01,
  -6.77824583E+01,-1.307927583E+02,-4.776693E-01,
  -9.876543210E+04,-9.393939111E+06,-9.876543210E+04,
  -9.876543210E+04,-9.393939111E+06,-9.876543210E+04,
  -9.876543210E+04,-9.393939111E+06,-9.876543210E+04

**Return Type**

Block data

**Default**

Not Applicable

---

**CALCulate:MEASurement:QAMPlitude?**

*(Read-Only)* Returns the current Amplitude Alignment (InstAlign) status.

**Relevant Modes**

SA

**Examples**

`CALC:MEAS:QAMP?`

**Return Type**

Boolean

0 - Alignment is current

1 - Alignment is questionable because Amplitude Alignment is in Hold or OFF.

See [:SENSe]:AMPLitude:ALIGnment[:STATe]

**Default**

Not Applicable

---

**:CALCulate:PARameter:COUNT <n>**

*(Read-Write)* Sets and returns the number of traces on the screen. All traces are displayed in separate windows. Use DISPlay:WINDow:SPLIT to set overlay (traces in same window) configurations.

Change the measurement parameter using `CALCulate:PARameter:DEFine`.

**Relevant Modes**

NA, NF

**Parameters**

<n> Number of traces.

---

Last Modified:

18-May-2011 Modified description text

28-Mar-2013 New command
For NA mode, choose from 1 to 4.
For NF mode, choose from 1 to 2.

**Examples**

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:PAR:COUN 2</td>
<td>Example use</td>
</tr>
</tbody>
</table>

**Default**

- `1`

Last Modified:

- **01june2018** Added NF mode.
- **15-Aug-2012** New command

---

**CALCulate:PARameter<tr>:DEFine <char>**

*(Read-Write)* Set and query the current measurement.

**Relevant Modes**

- CAT, NA, NF, VVM

**Parameters**

- `<tr>` Trace number (NA mode ONLY). Choose from 1 to 4. Choices 2, 3, 4 require that the appropriate multi-trace configuration first be created using :DISPlay:WNDow:SPLit.
- Trace number (NF mode Only). Choose from 1 and 2. Choice 2 requires that the appropriate multi-trace configuration first be created using :DISPlay:WNDow:SPLit.
- All other modes, do NOT specify.

- `<char>` Measurement parameter. Choose from:

  **For CAT Mode:**
  - RLOSs - Return Loss
  - DTF1 - Distance To Fault
  - DTF2 - DTF + Return Loss
  - DTF3 - DTF (VSWR)
  - DTF4 - DTF Linear
  - CLOSs - Cable loss 1 port
  - ILOSs - 2-port Insertion loss
  - VSWR - SWR
  - Available ONLY with Opt. 215
    - TDR - Linear Rho
    - STEP - TDR Ohm

  **For NA Mode:**
  - Reverse measurements are available ONLY with full S-parameter option.
    - S11 - Forward reflection measurement
• S21 - Forward transmission measurement
• S12 - Reverse transmission
• S22 - Reverse reflection
• A - A receiver measurement
• B - B receiver measurement
• R1 - Port 1 reference receiver measurement
• R2 - Port 2 reference receiver measurement

For NF Mode:
• NFIG - Noise Figure
• NFAC - Noise Factor
• GAIN - Gain
• NTEM - Noise temperature
• YFAC - Y-Factor

Available ONLY with Opt. 212
• SCC11 - Common reflect/common incident for logical port 1
• SDD11 - Differential reflect/differential incident for logical port 1
• SDC11 - Differential reflect/common incident for logical port 1.
• SCD11 - Common reflect/differential incident for logical port 1.

For VVM Mode:
• S11 - 1-port cable trimming
• S21 - 2-port transmission
• AB A/B ratio (NOT available on N9912A)
• BA B/A ratio (NOT available on N9912A)

Examples
CALC:PAR:DEF DTF2
'NA mode
CALC:PAR2:DEF S21

Query Syntax
CALCulate:PARameter<tr>:DEFine?

Return Type
Character

Default
Cat Mode: S11
NA Mode: S11 (trace 1)
VVM Mode:S11

Last Modified:
01-june-2018 Added Opt. 356 NF(10.3)
22-Jan-2015 Added Opt 215
24-Mar-2014 Added Opt 212
16-Jul-2013 Edited VVM mode choices
18-Oct-2012 Edited for new models
CALCulate:PARameter<n>:SELect

(Write-only)  Select (make active) the current trace. You can only select a displayed trace.
For CAT and NA, change the measurement parameter using CALCulate:PARameter:DEFine.

### Relevant Modes
CAT, NA, NF, and Pulse

### Parameters

- `<n>` Trace number.
  - For NA mode, choose from 1 to 4.
  - For CAT, NF, and Pulse modes, choose from 1 or 2.

### Examples
CALC:PAR2:SEL

### Query Syntax
Not Applicable

### Default
1

---

Last Modified:

- 01 June 2018  Added NF mode (10.3).
- 26 Apr-2012  Modified for CAT (5.75)

---

CALCulate:RELative[:MAGNitude]:AUTO <bool>

(Read-Write)  Set and query state of relative Power Meter measurements.

### Relevant Modes
Power Meter, Pulse Measurements, CPM

### Parameters

- `<bool>` Choose from:
  - OFF - Relative measurements OFF
  - ON - Relative measurements ON

### Examples
CALC:REL:AUTO 1

### Query Syntax
CALCulate:RELative[:MAGNitude]:AUTO?

### Return Type
Boolean

### Default
0

---

Last Modified:

- 1 Apr-2014  Added CPM
- 31 Oct-2013  Added Pulse
**:CALCulate:SPECTrum:MARKer:AOFF**

(Write-Only) Turns OFF all IQA spectrum markers.

**Relevant Modes**  
IQA (Spectrum only)

**Examples**  
CALC:SPEC:MARK:AOFF

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

**Last Modified:**

22oct2017  
Added IQA mode (10.1x)

**:CALCulate:SPECTrum:MARKer:CPSearch[:STATE] <bool>**

(Read-Write) Sets and queries the Continuous Peak Search. Enables or disables the Continuous Peak Search. When Continuous Peak Search is on, a peak search is automatically performed for the selected marker after each sweep. The rules for finding the peak are exactly the same as for Peak Search, including the use of peak criteria rules.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<boolean>ON (1) - Enables continuous peak search.  
OFF (0) - Disables continuous peak search.

**Couplings**  
N/A

**Examples**

CALC:SPEC:MARK:CPS ON  
CALC:SPEC:MARK:CPS

**Query Syntax**

CALC:SPEC:MARK:CPS?

**Default**  
OFF (0)

---

**Last Modified:**

22oct2017  
New IQA mode (10.1x)

**:CALCulate:SPECTrum:MARKer:DREF:FIXed <bool>**

(Read-Write) Set and query the state of fixed delta / reference markers.

**Relevant Modes**  
IQA (Spectrum only)
Parameters

<boolean>
ON (1) - Enables fixed delta reference markers.
OFF (0) - Disables fixed delta reference markers.

Couplings N/A

Examples

CALC:SPEC:MARK:DREF:FIX ON
CALC:SPEC:MARK:DREF:FIX 0

Query Syntax

CALC:SPEC:MARK:DREF:FIX?

Default

ON (1)

Last Modified:

22oct2017 New IQA mode (10.1x)

:CALCulate:SPECTrum:MARKer:FUNCtion:PEXCursion <num>

(Read-Only) Set and queries the minimum amplitude variation (rise and fall) required for a signal to be identified as a peak.

Relevant Modes IQA (Spectrum only)

Parameters

<num> Minimum: 0
Maximum: 200

Couplings For a signal to be identified as a peak it must meet certain criteria:
- Signals in the negative frequency range and signals very close to 0 Hz are ignored.
- Signal must satisfy peak excursion and peak threshold (:CALC:SPEC:MARK:FUNC:PTHR) criteria before being identified as a peak.

Examples

CALC:SPEC:MARK:FUNC:PEXC 5

Query Syntax

CALC:SPEC:MARK:FUNC:PEXC?

Default

6.00E+00

Last Modified:

22oct2017 New IQA mode (10.1x)

:CALCulate:SPECTrum:MARKer:FUNCtion:PTHReshold <num>

(Read-Write) Sets and queries the peak threshold value that defines the minimum signal level (or min
threshold) that the peak identification algorithm uses to recognize a peak.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<numeric>
Minimum: -200  
Maximum: 200

**Couplings**
For a signal to be identified as a peak it must meet certain criteria:
- Signals in the negative frequency range and signals very close to 0 Hz are ignored.
- Signal must satisfy peak excursion ([CALC:SPEC:MARK:FUNC:PEXC]) and peak threshold criteria before being identified as a peak.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:SPEC:MARK:FUNC:PTHR</td>
<td>-70</td>
</tr>
</tbody>
</table>

**Query Syntax**

CALC:SPEC:MARK:FUNC:PTHR?

**Default**

-9.000E+01

---

Last Modified:

22oct2017  
New IQA mode (10.1x)

---

:CALCulate:SPECTrum:MARKer[n]:FUNCtion <char>

(Read-Write) Causes the specified marker to become one of the IQA Marker functions. Also, enables or disables the specified marker.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<n>  
Existing marker to become a marker function. Choose from 1 to 6.

<character>  
Marker function. Choose from:
- **OFF** - Marker is returned to it's previous state (normal or delta).
- **NOISe** - Marker becomes a noise marker.
- **BPOWer** - Marker becomes a BandPower marker.

A Band Power marker integrates total power over the Band Power Span, which is set using:
- [CALC:SPEC:MARK:FUNC:BAND:SPAN]
- [CALC:SPEC:MARK:FUNC:BAND:SPAN:_AUTO]

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:SPEC:MARK1:FUNC</td>
<td>NOISe</td>
</tr>
</tbody>
</table>

**Query Syntax**

CALC:SPEC:MARK<n>:FUNC?

**Default**

**OFF**
**CALCulate:SPECTrum:MARKer:FUNCTION:BAND:SPAN <freq>**

*(Read-Write)* Set and read the frequency span for the selected I/Q marker.

Set **CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO** to OFF.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**  

*<frequency>* Frequency span value.

Couplings: If **CALCulate:MARKer:FUNCTION:BAND:SPAN:AUTO** is on, the value is 5\% of the existing frequency span.

**Examples**  
**CALC:SPEC:MARK:FUNC:BAND:SPAN 1e6**

**Query Syntax**  
**CALC:SPEC:MARK:FUNC:BAND:SPAN?**

**Default**  
500000

---

**CALCulate:SPECTrum:MARKer:FUNCTION:BAND:SPAN:AUTO**

*(Read-Write)* Set and read the method by which the frequency span for the selected I/Q marker.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**  

*<boolean>* Band power frequency span method.

**ON** (or 1) - Band Power frequency span is 5\% of existing frequency span.

**OFF** (or 0) - Set Band Power frequency span using **CALCulate:SPECTrum:MARKer:FUNCTION:BAND:SPAN**

**Examples**  
**CALC:SPEC:MARK:FUNC:BAND:SPAN:AUTO 1**

**Query Syntax**  
**CALCulate:SPECTrum:MARKer:FUNCTION:BAND:SPAN:AUTO?**

**Default**  
ON
:CALCulate:SPECtrum:MARKer<n>:FUNCtion:MAXimum

(Write-Only) Causes the specified marker to find the maximum amplitude of the trace.

**Relevant Modes**  IQA (Spectrum only)

**Parameters**

<n> New or existing marker to move. Choose from 1 to 6.

**Examples**  

CALC:SPEC:MARK3:FUNC:MAX

**Query Syntax**  N/A

**Default**  N/A

---

Last Modified:  

22oct2017  Added IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>:FUNCtion:MINimum

(Read-Only) Moves the selected marker to the minimum Y-axis value on the current trace. Minimum (negative) peak searches do not have to meet the peak search criteria. If the selected marker is OFF, it is turned ON before the minimum search is performed.

**Relevant Modes**  IQA (Spectrum only)

**Parameters**

<n> Existing marker to minimum (negative) peak. Choose from 1 to 6.

**Couplings**  N/A

**Examples**  

CALC:SPEC:MARK3:FUNC:MIN

**Query Syntax**  N/A

**Default**  N/A

---

Last Modified:  

22oct2017  New IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>:FUNCtion:PLEFt

(Read-Only) Moves the selected marker to the nearest peak left of the current marker which meets all enabled peak criteria.
If the selected marker was off, then it is turned on as a normal marker and a peak search is performed.

### Relevant Modes
IQA (Spectrum only)

### Parameters

<n> Existing marker to assign to next highest peak. Choose from 1 to 6.

### Couplings
N/A

### Examples
CALC : SPEC : MARK<3> : FUNC : PLEF

### Query Syntax
N/A

### Default
N/A

Last Modified:
22oct2017 New IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>:FUNCtion:PRIGhT

(Write-Only) Moves the selected marker to the nearest peak right of the current marker which meets all enabled peak criteria.

If the selected marker was off, then it is turned on as a normal marker and a peak search is performed.

### Relevant Modes
IQA (Spectrum only)

### Parameters

<n> Existing marker to assign to next highest peak. Choose from 1 to 6.

### Couplings
N/A

### Examples
CALC : SPEC : MARK<3> : FUNC : PRIG

### Query Syntax
N/A

### Default
N/A

Last Modified:
22oct2017 New IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>:PHASe?

(Read-Only) Read the frequency phase for the selected IQA marker. The unit is in radians and will be a
value between ±π or ±3.14. If the marker is inactive or outside the range of the trace, a value of -9.9E+37 is returned.

If the specified marker is a delta marker, this query will return the phase delta from the reference marker.

**Relevant Modes**  IQA (Spectrum only)

**Parameters**

<n>  Existing marker to assign a frequency span. Choose from 1 to 6.

Couplings  Marker needs to be active, on the spectrum trace and within the current range of the spectrum trace.

Resolution  0.01 radians

**Examples**  

CALC:SPEC:MARK<n>:PHAS?

**Query Syntax**  

CALC:SPEC:MARK<n>:PHAS?

**Default**  

Returns 9.91E+37 if marker inactive or on the IQ trace.

Last Modified:

22oct2017  New IQA mode (10.1x)

---

:**CALCulate:SPECTrum:MARKer<n>:SET:CENTer**

(Write-Only) Sets the center frequency to the selected marker. The center frequency moves to the current selected marker's position at the center of the display.

**Relevant Modes**  IQA (Spectrum only)

**Parameters**

<n>  Existing marker that is used to set the center frequency. Choose from 1 to 6.

Couplings  N/A

**Examples**  

CALC:SPEC:MARK<n>:SET:CENT

**Query Syntax**  N/A

**Default**  N/A

Last Modified:

22oct2017  New IQA mode (10.1x)
(Read-Only) Sets the reference level to the amplitude value of the selected marker, moving the marked point to the reference level (top line of the graticule).

If the currently selected marker OFF when this control is pressed, it will be turned ON at the center of the screen as a normal type marker, and its amplitude applied to the reference level.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<n> Existing marker that is used to set the reference level. Choose from 1 to 6.

**Couplings**

N/A

**Examples**

CALC:SPEC:MARK3:SET:RLEV

**Query Syntax**

N/A

**Default**

N/A

Last Modified:

22oct2017  
New IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer[n]:TRACe <int>

(Write-Read) Moves an existing marker to the specified trace number.

**Note:** This feature is called Marker Trace in the User's Guide.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<n> Existing marker to assign to trace. Choose from 1 to 6.

<tracenumber> Trace number. Choose from:

IQA:

- 1, 2, 3, 4

**Examples**

CALC:SPEC:MARK1:TRAC 1

**Query Syntax**

CALC:SPEC:MARK<n>:TRAC?

**Return Type**

Integer

**Default**

1

Last Modified:

22oct2017  
Added IQA mode (10.1x)
:CALCulate:SPECtrum:MARKer<n>:X <num>

(Read-Write) Set and query the X-axis location for the specified marker. See To Create and Move a Delta Marker.

**Relevant Modes**  IQA (Spectrum only)

**Parameters**

<n>  Existing marker for which to set X-axis location. Choose from 1 to 6.

<number>  X-axis location. Choose any frequency value currently displayed on the X-axis.

Minimum: -9.9E+37 Hz

Maximum: 9.9E+37 Hz

**Dependencies**

Range is dependent on the X axis range of the selected trace.

**Examples**  CALC:SPEC:MARK1:X 4e9

**Query Syntax**  CALC:SPEC:MARK<n>:X?

**Return Type**  Numeric

**Default**  OFF.

---

Last Modified:

22oct2017  Added IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>:Y?

(Read-Only) Reads the Y-axis value for the specified marker.

**Relevant Modes**  IQA (Spectrum only) - One value is returned: Magnitude

**Parameters**

<n>  Existing marker for which to read Y-axis value. Choose from 1 to 6.

**Dependencies**

Range is dependent on the X axis range of the selected trace.

**Examples**  CALC:SPEC:MARK3:Y?

**Return Type**  Numeric

**Default**  OFF.

---

Last Modified:

22oct2017  Added IQA mode (10.1x)

---

:CALCulate:SPECtrum:MARKer<n>[[:STATe] <char>
(Read-Write) Create, change, or remove a marker.

**Relevant Modes**  
IQA (Spectrum only)

**Parameters**

<n> New or existing marker to create, change, or remove. Choose from 1 to 6.

<character> Choose from:

-NORM - Marker is a Normal marker
-DELT - Marker is a Delta marker pair.
-OFF - Marker is disabled.

**To create and move a delta marker:**

1. Create a normal marker using this command.
2. Move the marker to the reference position using `CALC:SPEC:MARK[n]:X <num>`
3. Change the marker to a delta marker using this command.
4. Move the delta marker to the delta position using `CALC:SPEC:MARKer[n]:X <num>`

**Examples**

```
CALC:SPEC:MARK1 NORM
```

**Query Syntax**

`CALC:SPEC:MARKer<n>?`

**Return Type** Character

**Default** OFF

---

Last Modified:  
22oct2017 Added IQA mode (10.1x)

---

`:CALCulate:WAVeform:MARKer:AOFF`

(Write-Only) Turns OFF all IQA waveform markers.

**Relevant Modes**  
IQA (Waveform only)

**Examples**

```
CALC:WAV:MARK:AOFF
```

**Query Syntax** Not Applicable

**Default** Not Applicable

---

Last Modified:  
22oct2017 Added IQA mode (10.1x)
### :CALCulate:WAVEform:MARKer:COUPle:X <num>

**Command**: Set the X-axis coupled marker. Sets the X-axis value to the value of the other coupled markers.

**Parameters**
- `<numeric>`  
  - **Minimum**: -9.9e+37  
  - **Maximum**: 9.9e+37

**Examples**
- `CALC:WAV:MARK:COUP 20e-6`

**Relevant Modes**  
IQA (Waveform Only)

**Default**  
n/a

**Last Modified**  
21nov2017  
Added IQA mode (10.1x)

### :CALCulate:WAVEform:MARKer:COUPle[:STATe] <bool>

**Command**: Sets or reads the state of the coupled markers. If a selected marker's marker type is Normal/Delta, all active markers' X value will be coupled to selected marker's value.

If selected marker's marker type is OFF, all active markers X value still keep their own value.

**Parameters**
- `<boolean>`  
  - **ON** (1) - Enables coupled markers.  
  - **OFF** (0) - Disables coupled markers.

**Examples**  
N/A

**Query Syntax**
- `CALC:WAV:MARK:COUP ON`
- `CALC:WAV:MARK:COUP 0`

**Return Type**
- `CALC:WAV:MARK:COUP?`

**Default**  
OFF (0)

**Last Modified**  
22oct2017  
Added IQA mode (10.1x)
CALC:WAV:MARK:CPSearch[:STATe] <bool>

(Read-Write) Sets and read the Continuous Peak Search. Enables or disables the Continuous Peak Search. When Continuous Peak Search is on, a peak search is automatically performed for the selected marker after each sweep. The rules for finding the peak are exactly the same as for Peak Search, including the use of peak criteria rules.

**Relevant Modes**  
IQA (Waveform Only)

**Parameters**

<boolean>

ON (1) - Enables continuous peak search.

OFF (0) - Disables continuous peak search.

**Couplings**

N/A

**Examples**

CALC:WAV:MARK:CPS ON

CALC:WAV:MARK:CPS 0

**Query Syntax**

CALC:WAV:MARK:CPS?

**Default**

OFF (0)

---

Last Modified:

22oct2017  
New IQA mode (10.1x)

---

CALC:WAV:MARK:DFixed <bool>

(Read-Write) Set and read the state of fixed delta / reference markers.

**Relevant Modes**  
IQA (Waveform Only)

**Parameters**

<boolean>

ON (1) - Enables fixed delta reference markers.

OFF (0) - Disables fixed delta reference markers.

**Couplings**

N/A

**Examples**

CALC:WAV:MARK:DREF:FIX ON

CALC:WAV:MARK:DREF:FIX 0

**Query Syntax**

CALC:WAV:MARK:DREF:FIX?

**Default**

ON (1)

---

Last Modified:
:CALCulate:WAVeform:MARKer:FUNCtion:PEXCursion <num>

(Read-Write) Set and queries the minimum value variation (rise and fall) required for a signal to be identified as a peak.

**Relevant Modes**  IQA (Waveform Only)

**Parameters**

<numeric>
Minimum: 0
Maximum: 200

**Couplings**

For a signal to be identified as a peak it must meet certain criteria:
- Signals in the negative frequency range and signals very close to 0 Hz are ignored.
- Signal must satisfy peak excursion and peak threshold (:CALC:WAV:MARK:FUNC:PTHR) criteria before being identified as a peak.

**Examples**

CALC:WAV:MARK:FUNC:PEXC 2

**Query Syntax**

CALC:WAV:MARK:FUNC:PEXC?

**Default**

0.00E+00

---

:CALCulate:WAVeform:MARKer:FUNCtion:PTHReshold <num>

(Read-Write) Set and queries the peak threshold value that defines the minimum signal level (or min threshold) that the peak identification algorithm uses to recognize a peak.

**Relevant Modes**  IQA (Waveform Only)

**Parameters**

<numeric>
Minimum: -200
Maximum: 200

**Couplings**

For a signal to be identified as a peak it must meet certain criteria:
- Signals in the negative value range and signals very close to 0 Hz are ignored.
- Signal must satisfy peak excursion (:CALC:WAV:MARK:FUNC:PEXC) and peak threshold criteria before being identified as a peak.

**Examples**

CALC:SPEC:WAV:FUNC:PTHR -20

**Query Syntax**

CALC:SPEC:WAV:FUNC:PTHR?
Default: -9.000E+01

Last Modified:
22oct2017 New IQA mode (10.1x)

:CALCulate:WAVeform:MARKer<n>:DATA <char>
(Read-Write) Set and read the results of the waveform trace that a marker has been assigned.
This command operates on the selected trace. First select a trace using TRAC:WAV<n>:TYPE.

Relevant Modes: IQA (Waveform only)

Parameters:
- <n> Existing marker
- <char> Marker function:
  - WAV: Marker is real
  - POL: Marker is polar
  - PHAS: Marker is complex
  - UPH: Marker is upper
  - REAL: Marker is real
  - IMAG: Marker is imaginary

Examples:
- CALC:WAV:MAR
- CALC:WAV:MAR

Query Syntax: CALC:WAV:MAR

Return Type: character

Default: WAV

Last Modified:
22oct2017 Added IQA mode (10.1x)

:CALCulate:WAVeform:MARKer<n>:FUNCtion <char>
(Read-Write) Causes the specified marker to become one of the IQA Marker functions. Also, enables or disables the specified marker.

Relevant Modes: IQA (Waveform only)

Parameters

Existing marker to become a marker function. Choose from 1 to 6.

Marker function. Choose from:

- **OFF**: Marker is returned to its previous state (normal or delta).
- **NOISe**: Marker becomes a noise marker.
- **BPOWER**: Marker becomes a Band/Interval Power marker.

An Interval Power marker calculates the average power over the Interval Power Span, which is set using:

- `CALC:WAV:MARK:FUNC:INT:SPAN`

<table>
<thead>
<tr>
<th>Examples</th>
<th>Query Syntax</th>
<th>Default</th>
</tr>
</thead>
</table>

---

Set and read the waveform maker span for the selected IQA marker.

**Relevant Modes**: IQA

**Parameters**

- **<numeric>**: Assign a waveform span.

If `CALC:WAV:MARK:FUNC:INT:SPAN:AUTO` is on, the value is 5% of the existing frequency span.

<table>
<thead>
<tr>
<th>Examples</th>
<th>Query Syntax</th>
<th>Default</th>
</tr>
</thead>
</table>

---

Last Modified:

22oct2017 New IQA mode (10.1x)
(Read-Write) Set and read the automatic interval span setting for the selected I/Q marker. See also `CALCulate:WAVEform:MARKer:FUNCTION:INTerval:SPAN`.

**Relevant Modes** IQA (Waveform only)

**Parameters**

- `<bool>` Band power frequency span method.
  - **ON** (or 1) - Interval Power frequency span is 5% (default) of existing frequency span (Stop Time - Start Time).
  - **OFF** (or 0) - Set Interval Power frequency span manually using `CALCulate:WAVEform:MARKer:FUNCTION:INTerval:SPAN`

**Examples**

```
CALC:WAV:MARK:FUNC:INT:SPAN:AUTO 1
```

**Query Syntax** `CALCulate:WAVEform:MARKer:FUNCTION:INT:SPAN:AUTO?`

**Default** ON

---

Last Modified: 22oct2017 New IQA mode (10.1x)

---

`:CALCulate:WAVEform:MARKer<n>:FUNCTION:MAXimum`

(Write-Only) Causes the specified marker to find the maximum amplitude of the trace. If the marker is off it activated and is set to NORMal.

**Relevant Modes** IQA (Waveform only)

**Parameters**

- `<n>` New or existing marker to move. Choose from 1 to 6.

**Examples**

```
CALC:WAV:MARK1:FUNC:MAX
```

**Query Syntax** N/A

**Default** N/A

---

Last Modified: 22oct2017 Added IQA mode (10.1x)

---

`:CALCulate[:SELected]:MARKer:FUNCTION:Mn:MIN`
(Write-only) Sets the frequency of the frequency counter marker in Hz. See also CALC:MARK:FUNC:Mn:MAX.

Note:
This SCPI command sets either marker 5 to a minimum value between marker 1 and 2 or it sets marker 6 to a minimum value between marker 3 and 4 (i.e., CALC:MARK:FUNC:Mn:MIN, where n=5 or n=6). No other marker choices are possible.

**For marker 5 search:** If marker 1, 2, 5 are OFF or Delta when MIN is initiated:
- Markers are all changed to Normal
- Marker 1 is set to 1/4 of full span
- Marker 2 is set to 1/4 of full span

**For marker 6 search:** If marker 3, 4, 6 are OFF or Delta when MIN is initiated:
- Markers are all changed to Normal
- Marker 3 is set to 1/4 of full span
- Marker 4 is set to 1/4 of full span

Marker 1, 2, 3, and 4 positions are not modified, if they have been previously positioned prior to initiating this command.

**Relevant Modes** CAT/TDR

**Parameters** None

**Examples**
- Inserts Marker 5 at the next minimum between Markers 1 and 2:
  
  `CALC:MARK:FUNC:M5:MIN`

- Inserts Marker 6 at the next minimum between Markers 3 and 4:
  
  `CALC:MARK:FUNC:M6:MIN`

**Return Type** n/a

**Default** n/a

---

:CALCulate:WAVEform:MARKer<n>:FUNCtion:PNEXt

(Write-Only) Reads the selected waveform marker. Sets the selected marker to the peak that has the next highest (but, lesser value) amplitude from the marker’s current value. Only peaks which meet all enabled peak criteria are considered.

If the selected marker was off, then it is turned on as a normal marker and a peak search is performed.

**Relevant Modes** IQA (Waveform Only)

**Parameters**

- `<n>` Existing marker to assign to next highest peak. Choose from 1 to 6.

- Couplings N/A
**:CALCulate:WAVeform:MARKer<n>:TRACe <int>**

(Write-Read) Moves an existing marker to the specified trace number.

**Note:** This feature is called Marker Trace in the User's Guide.

**Relevant Modes**  
IQA (Waveform only)

**Parameters**

- `<n>`  
  Existing marker to assign to trace. Choose from 1 to 6.

- `<trace number>`  
  Trace number. Choose from:
  - IQA:
    - 1, 2, 3, 4

**Examples**

- `CALC:WAV:MARK1:TRAC 3`
- `CALC:WAV:MARK2:TRAC 2`

**Query Syntax**  
`CALC:WAV:MARK<n>:TRAC?`

**Default**  
1

Last Modified:

22oct2017  
Added IQA mode (10.1x)

---

**:CALCulate:WAVeform:MARKer<n>:X**

(Read-Write) Set and query the X-axis location for the specified waveform marker. No effect if the marker is Off. See [To Create and Move a Delta Marker](#).

**Relevant Modes**  
IQA (Waveform only)

**Parameters**

- `<n>`  
  Marker to set on the X-axis location. Choose from 1 to 6.

- `<numeric>`  
  X-axis location. Choose any value currently displayed on the X-axis.
  Minimum: -9.9E+37 Hz
Maximum: $9.9\times10^{37}$ Hz

Dependencies
Range is dependent on the X axis range of the selected trace.

Examples
CALC:WAV:MARK3:X \(80e-6\)

Query Syntax
CALC:WAV:MARK<n>:X?

Return Type
Numeric

Default
OFF

---

Last Modified:
22oct2017 Added IQA mode (10.1x)

---

**CALCulate:WAVeform:MARKer<n>:Y?**

(Read-Only) Reads the Y-axis value for the specified marker. One value is returned: Magnitude

**Relevant Modes**
IQA (Waveform only)

**Parameters**

\(<n>\) Existing marker for which to read Y-axis value. Choose from 1 to 6.

**Dependencies**

Examples
CALC:WAV:MARK1:Y?

**Return Type**
numeric

**Default**
OFF

---

Last Modified:
22oct2017 Added IQA mode (10.1x)

---

**CALCulate:WAVeform:MARKer<n>[:STATe]**

(Read-Write) Set or query the marker control mode. Sets the selected marker to Normal, Delta or Off. If all markers are Off, setting a Marker sets the selected marker to Normal and places it at the center of the screen on the trace determined by the Marker Trace rules. Marker X Axis Value is displayed.

**Relevant Modes**
IQA (Waveform only)

**Parameters**

\(<n>\) New or existing marker to create, change, or remove. Choose from 1 to 6.

\(<\text{character}>\) Choose from:

NORM - Marker is a Normal marker

---

170
DELT - Marker is a Delta marker pair.
OFF - Marker is disabled.

To create and move a delta marker:
1. Create a normal marker using this command.
2. Move the marker to the reference position using 
   \[ \text{CALC:WAV:MARK[n]:X <time>} \]
3. Change the marker to a delta marker using this command.
4. Move the delta marker to the delta position using
   \[ \text{CALC:WAV:MARK[n]:X <time>} \]

Examples

\[ \text{CALC:WAV:MARK2 NORM} \]
\[ \text{CALC:WAV:MARK2 DELT} \]

Query Syntax

\[ \text{CALC:WAV:MARK<n>?} \]

Return Type

Character

Default

Off

Last Modified:
22oct2017 Added IQA mode (10.1x)

:DISPlay:TABLe:MARKer:SPECTrum <bool>

(Read-Write) Set and query the display of the marker table.

Relevant Modes

IQA (Spectrum only)

Parameters

<boolean> Marker table display state. Choose from:
  O or OFF - Table OFF
  1 or ON - Table ON

CoupingsIf marker table is enabled, the marker table is only displayed when the active window number is 1.

Examples

\[ \text{DISP:TABL:MARK:SPEC ON} \]

Query Syntax

\[ \text{DISPlay:TABLe:MARKer:SPECTrum?} \]

Return Type

Boolean

Default

OFF

Last modified:
22oct2017 Added IQA mode (10.1x)
:DISPlay:TABLE:MARker:WAVeform <bool>

(Read-Write) Set or read the marker table. Enables the marker table to display below the graph on the FieldFox.

**Relevant Modes**  IQA (Waveform Only)

**Parameters**

<boolean>  Band power frequency span method.

**ON** (or 1) - Enables the marker table below the graph that displays the status of all six of the waveform markers.

**OFF** (or 0) - Disables the displayed marker table.

**Couplings**

If marker table is enabled, the marker table is only displayed when the active window number is 1.

**Examples**

```
DISP:TABL:MARK:WAV 1
```

**Query Syntax**

```
DISP:TABLE:MARK:WAV?
```

**Return Type**  Boolean

**Default**  OFF

Last Modified:

22oct2017  New IQA mode (10.1x)

---

CALCulate[:SELected]:AMPLitude:MARKer:DELTa:STATe <bool>

(Read-Write)  Set or query the state of the Delta Amplitude Markers. To be used as Delta markers, Amplitude Markers must first be enabled using `CALCulate[:SELected]:AMPLitude:MARKer:STATe`.

**Relevant Modes**  Pulse Measurements, FOPS

**Parameters**

<bool>  Choose from:

**OFF or 0** - Delta Amplitude markers OFF

**ON or 1** - Delta Amplitude markers OFF

**Examples**

```
CALC:AMPL:MARK:DELT:STAT 1
```

**Query Syntax**

```
CALCulate[:SELected]:AMPLitude:MARKer:DELTa:STATe?
```

**Return Type**  Boolean

**Default**  Off

Last Modified:
CALCulate[:SELected]:AMPLitude:MARKer:STATe <bool>

(Read-Write) Set or query the state of the two Amplitude Markers.

Relevant Modes  Pulse Measurements,  FOPS

Parameters

<bool> Choose from:
OFF or 0 - Amplitude markers OFF
ON or 1 - Amplitude markers OFF

Examples  CALC:AMPL:MARK:STAT 1

Query Syntax  CALCulate[:SELected]:AMPLitude:MARKer:STATe?

Return Type  Boolean

Default  Off

Last Modified:
21-Mar-2014  Added FOPS
29-Oct-2013  New command

CALCulate[:SELected]:AMPLitude:MARKer:Y1 <num>

(Read-Write) Set or query the Y-axis location of the Amplitude Marker 1.

Relevant Modes  Pulse Measurements,  FOPS

Parameters

<num> Y-axis location in dBm.

Examples  CALC:AMPL:MARK:Y1 -1.5

Query Syntax  CALCulate[:SELected]:AMPLitude:MARKer:Y1?

Return Type  Numeric

Default  0

Last Modified:
21-Mar-2014  Added FOPS
29-Oct-2013  New command
CALCulate[:SELected]:AMPLitude:MARKer:Y2 <num>

(Read-Write) Set or query the Y-axis location of the Amplitude Marker 2.

Relevant Modes  Pulse Measurements, FOPS

Parameters

| <num> | Y-axis location in dBm. |

Examples  CALC:AMPL:MARK:Y2 -1.5

Query Syntax  CALCulate[:SELected]:AMPLitude:MARKer:Y2?

Return Type  Numeric

Default  0

Last Modified:

21-Mar-2014  Added FOPS
29-Oct-2013  New command

CALCulate[:SELected]:CONVersion:FUNCtion <char>

(Read-Write) Set and query measurement conversion function for the selected measurement. First select a measurement using CALCulate:PARameter:SElECT.

Relevant Modes  NA

Parameters

<table>
<thead>
<tr>
<th>&lt;char&gt;</th>
<th>Conversion function. Choose from:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>No conversion</td>
</tr>
<tr>
<td>ZAUTO</td>
<td>The displayed S-parameter is converted to the appropriate Z parameter: Refl for S11 and S22; Trans for S21 and S12.</td>
</tr>
<tr>
<td>YAUTO</td>
<td>The displayed S-parameter is converted to the appropriate Y parameter: Refl for S11 and S22; Trans for S21 and S12.</td>
</tr>
<tr>
<td>ZREFlection</td>
<td>The displayed S-parameter is converted to Z reflection, regardless of whether the S-parameter is reflection (S11 or S22) or transmission (S21 or S12).</td>
</tr>
<tr>
<td>YREFlection</td>
<td>The displayed S-parameter is converted to Y reflection, regardless of whether the S-parameter is reflection (S11 or S22) or transmission (S21 or S12).</td>
</tr>
<tr>
<td>ZTRANSmission</td>
<td>The displayed S-parameter is converted to Z transmission, regardless of whether the S-parameter is reflection (S11 or S22) or transmission (S21 or S12).</td>
</tr>
<tr>
<td>YTRANSmission</td>
<td>The displayed S-parameter is converted to Y transmission, regardless of whether the S-parameter is reflection (S11 or S22) or transmission (S21 or S12).</td>
</tr>
<tr>
<td>INVerse</td>
<td>The displayed S-parameter is converted to 1/S-parameter.</td>
</tr>
</tbody>
</table>

Examples  CALC:CONV:FUNC ZAUT

174
### CALCulate[:SELected]:CORRection:EDELay:TIME <num>

(Read-Write) Set and query the electrical delay for the selected trace. Use `CALCulate:PARameter:SELect` to select a trace.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;num&gt; Electrical Delay in seconds. Choose a value between 0 and 10 seconds.</td>
</tr>
<tr>
<td>Examples</td>
<td><code>CALC:COR:EDEL:TIME 5e-10</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td><code>CALCulate[:SELected]:CORRection:EDELay:TIME?</code></td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>

Last Modified:

25-Mar-2014 New command (A.07.50)

### CALCulate[:SELected]:CORRection:OFFSet:PHASe <num>

(Read-Write) Set and query the phase offset for the selected trace. Use `CALCulate:PARameter:SELect` to select a trace.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;num&gt; Phase offset in degrees. Choose a value between 0 and 360 degrees.</td>
</tr>
<tr>
<td>Examples</td>
<td><code>CALC:COR:OFFS:PHAS 20</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td><code>CALCulate[:SELected]:CORRection:OFFSet:PHASe?</code></td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>

Last Modified:

19-Oct-2010 New command (5.30)
CALCulate[:SELected]:DATA:FDATa <data>

(Write-Read) Send and read the selected trace data in the current display format - one value per data point. Undefined behavior for Smith or Polar formats. (i.e., this is a screen capture from the currently displayed trace.)
Select a trace with CALCulate:PARameter:SELect
Set format with CALCulate:FORMat

Relevant Modes  NA
CAT - Read-only

Parameters
<data>  Comma-separated data to send.

Examples  
'send three data points
CALC:DATA:FDAT 1,1,1

Query Syntax  CALCulate[:SELected]:DATA:FDATa?
If correction is ON, then the returned data is corrected.

Return Type  Comma-separated numeric
Default  Not Applicable

CALCulate[:SELected]:DATA:FMEM <data>

(Write-Read) Send and read the selected memory trace data in the current display format - one value per data point.
Select a trace with :CALCulate:PARameter:SELect
Set format with CALCulate:FORMat

Relevant Modes  NA

Parameters
<data>  Comma-separated data to send.

Examples  
'send three data points
CALC:DATA:FMEM 1,1,1

Query Syntax  CALCulate[:SELected]:DATA:FMEM?
A memory trace must first be stored using CALCulate:MATH:MEMorize
If correction is ON, then the returned data is corrected.

Return Type  Comma-separated numeric
Default  Not Applicable
CALCulate[SELected]:DATA:SDATA <data>

(Write-Read) Send and read the selected trace data - unformatted - two values per data point (Real, Imaginary). (i.e., includes any error correction that has been applied to the trace data.)
Select a trace with :CALCulate:PARameter:SELection

Relevant Modes NA

Parameters

<data> Comma-separated data to send.

Examples send three data points
CALC:DATA:SDAT 1,1,1

Query Syntax CALCulate[SELected]:DATA:SDATA?
If correction is ON, then the returned data is corrected.
If imaginary data is not available, 0 is returned.

Return Type Comma-separated numeric

Default Not Applicable

CALCulate[SELected]:DATA:SMEM <data>

(Write-Read) Send and read the selected memory trace data - unformatted - two value per data point (Real, Imaginary).
Select a trace with :CALCulate:PARameter:SELection

Relevant Modes NA

Parameters

<data> Comma-separated data to send.

Examples send three data points
CALC:DATA:SMEM 1,1,1

Query Syntax CALCulate[SELected]:DATA:SMEM?
A memory trace must first be stored using CALCulate:MATH:MEMorize
If correction is ON, then the returned data is corrected.
If imaginary data is not available, 0 is returned.

Return Type Comma-separated numeric

Default Not Applicable

CALCulate[SELected]:FILTER[GLE]:TIME:CENTer <num>

(Read-Write) Set and query the gate filter center time.

Relevant Modes NA
**Parameters**

<num>  Center time in seconds. Choose any number between: ± \((\text{number of points}-1) / \text{frequency span}\).

**Examples**

CALC:FILT:TIME:CENT 1e-9

**Query Syntax**

CALCulate[:SELected]:FILTer[:GATE]:TIME:CENTer?

**Return Type**

Numeric

**Default**

0

---

**CALCulate[:SELected]:FILTer[:GATE]:TIME:SHAPe <char>**

*(Read-Write)* Set and query the gating filter shape.

**Relevant Modes**  NA

**Parameters**

<char>  Choose from

\text{MAXimum} - the widest gate filter available

WIDE -

\text{NORMal} -

\text{MINimum} - the narrowest gate filter available

**Examples**

CALC:FILT:TIME:SHAP NORM

**Query Syntax**

CALCulate[:SELected]:FILTer[:GATE]:TIME:SHApe?

**Return Type**

Character

**Default**

NORMal

---

**CALCulate[:SELected]:FILTer[:GATE]:TIME:SPAN <num>**

*(Read-Write)* Set and query the gate filter span time.

**Relevant Modes**  NA

**Parameters**

<num>  Span time in seconds. Choose any number between: 0 and \(2^*\ [(\text{number of points}-1) / \text{frequency span}]\)
**CALCulate[:SELected]:FILTer[:GATE]:TIME:SPAN <num>**

**Example**
CALC:FILT:TIME:SPAN 5ns

**Query Syntax**
CALCulate[:SELected]:FILTer[:GATE]:TIME:SPAN?

**Return Type**
Numeric

**Default**
20 ns

Last Modified:
24-Jan-2012 New command

**CALCulate[:SELected]:FILTer[:GATE]:TIME:STARt <num>**

(Read-Write) Set and query the gate filter start time.

**Relevant Modes**
NA

**Parameters**

<num> Start time in seconds. Choose any number between:
± (number of points-1) / frequency span

**Examples**
CALC:FILT:TIME:STARt 5ns

**Query Syntax**
CALCulate[:SELected]:FILTer[:GATE]:TIME:STARt?

**Return Type**
Numeric

**Default**
10 ns

Last Modified:
24-Jan-2012 New command

**CALCulate[:SELected]:FILTer[:GATE]:TIME:STATe <bool>**

(Read-Write) Set and query the ON | OFF gating state for the active trace. Select a trace using CALCulate:PARameter:SELect.

**Relevant Modes**
NA

**Parameters**

<bool> Gating state. Choose from:
0 or OFF - Gating is OFF
1 or ON - Gating is ON

**Examples**
CALC:FILT:TIME:STAT 1

**Query Syntax**
CALCulate[:SELected]:FILTer[:GATE]:TIME:STATe?
CALCulate[:SELected]:FILTer[:GATE]:TIME:STOP <num>

(Read-Write) Set and query the gate filter Stop time.

**Parameters**

<num> Stop time in seconds. Choose any number between:

\[ \pm \frac{\text{(number of points}-1)}{\text{frequency span}} \]

**Examples**

CALC:FILT:TIME:STOP 5ns

**Query Syntax**

CALCulate[:SELected]:FILTer[:GATE]:TIME:STOP?

**Return Type** Numeric

**Default** 10 ns

Last Modified:

24-Jan-2012 New command

---

CALCulate[:SELected]:FILTer[:GATE]:TIME[:TYPE] <char>

(Read-Write) Set and query the gate filter type.

**Parameters**

<char> Choose from:

- **BPASs** - Includes (passes) the range between the start and stop times.
- **NOTCh** - Excludes (attenuates) the range between the start and stop times.

**Examples**

CALC:FILT:TIME BPAS

**Query Syntax**

CALCulate[:SELected]:FILTer[:GATE]:TIME[:TYPE]?

**Return Type** Character

**Default** BPAS

Last Modified:

24-Jan-2012 New command
### CALCulate:[SElected]:FMEM:DATA?

(Read-only) Read the data from a memory trace. A memory trace must first be stored using CALCulate[:SElected]:MATH:MEMorize.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>FOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Examples</td>
<td>CALC:FMEM:DATA?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Last Modified:

21-Mar-2014 New command

### CALCulate[:SElected]:FORMat <char>

(Read-Write) Set and query displayed data format of the NA mode measurement.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;char&gt; Data format. Choose from:</td>
</tr>
<tr>
<td></td>
<td>MLOGarithmic – Log magnitude</td>
</tr>
<tr>
<td></td>
<td>MLIinear – Linear magnitude</td>
</tr>
<tr>
<td></td>
<td>SWR – Standing Wave Ratio</td>
</tr>
<tr>
<td></td>
<td>PHASE - Phase in degrees. The trace wraps every 360 degrees, from +180 to –180</td>
</tr>
<tr>
<td></td>
<td>UPHase - Unwrapped phase in degrees.</td>
</tr>
<tr>
<td></td>
<td>SMITH – Smith chart; series resistance and reactance.</td>
</tr>
<tr>
<td></td>
<td>POLar - Magnitude and phase of the reflection coefficient.</td>
</tr>
<tr>
<td></td>
<td>GDELeay – Group delay (N9912A - S11 ONLY)</td>
</tr>
<tr>
<td></td>
<td>ZMAG - Impedance, magnitude only.</td>
</tr>
<tr>
<td></td>
<td>REAL - Resistive portion of the measured complex data.</td>
</tr>
<tr>
<td></td>
<td>IMAGinary - Reactive portion of the measured data.</td>
</tr>
<tr>
<td>Examples</td>
<td>CALC:FORMat SWR</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>CALCulate[:SElected]:FORMat?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Character</td>
</tr>
</tbody>
</table>
CALCulate[:SELected]:GAIN:DATA??

(Read-Only) Return an array of floating point values, representing Gain. (Data – Memory).

Relevant Modes  FOPS

Parameters

Examples  CALC:GAIN:DATA?

Return Type  Numeric

Default  Not Applicable

Last Modified:

19-Dec-2018  New FOPS command

CALCulate[:SELected]:LIMit:BEEP <bool> - Superseded

This command is replaced with CALCulate[:SELected]:LIMit:SOUNd. Learn about superseded commands.

(Read-Write) Set and query whether the FieldFox beeps when a limit line failure occurs.

For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELect.

Relevant Modes  CAT, NA, SA

Parameters

Examples  CALC:LIM:BEEP ON

Query Syntax  CALCulate[:SELected]:LIMit:BEEP?

Return Type  Boolean

Default  OFF

Last Modified:
CALCulate[:SELected]:LIMit:DATA <data> - Superseded

This command is replaced with CALCulate:LIMit:LLData which can also set Relative Limits.

(Read-Write) Set and query the data to complete the limit line table.

For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELection.


Relevant Modes SA, CAT, NA

Parameters

- <data> Data for all limit segments in the following format:
  - n = number of segments, followed by segment data.
  - each segment: State, Type, BegStim, EndStim, BegResp, EndResp
  - Where:
    - State 0 for limit line disabled
      1 for limit line enabled.
    - Type Type of limit segment. Choose from
      0 - Upper limit
      1 - Lower limit
    - BegStim Start of X-axis value (freq, power, time)
    - EndStim End of X-axis value
    - BegResp Y-axis value that corresponds with Start of X-axis value
    - EndResp Y-axis value that corresponds with End of X-axis value
  - Subsequent segments are appended to the data in the same manner.

Examples

- The following writes three upper limit segments for a 40 MHz bandpass filter.
  - individual segments are highlighted for readability.
  - CALC:LIM:DATA 3,1,0,2e7,3e7,-
  - 30,0,1,0,3e7,5e7,0,0,1,0,5e7,6e7,0,-30

Query Syntax CALCulate:LIMit:DATA?

Return Type Block data

Default 0 - Limit line data off
This command replaces \texttt{CALCulate:LIMit:DATA} which can be used ONLY with Fixed limit lines. This command can also be used with Relative Limit Lines.

(Read-Write) Set and query the data to complete the limit line table.

For NA and CAT mode, this command operates on the selected trace. First select a trace using \texttt{CALC:PAR<tr>:SELection}.

For ACP measurements, use \texttt{CALCulate:ACPower:OFFSet:LIST:LIMit:NEGative[:UPPer]:DATA} and \texttt{CALCulate:ACPower:OFFSet:LIST:LIMit:POSitive[:UPPer]:DATA}.

**Relevant Modes** SA, CAT, NA, NF

**Parameters**

\texttt{<data>} Data for all limit segments in the following format:

- $n =$ number of segments, followed by segment data.
- Where:

  - $nL =$ number of limit line segments to follow
  - \textbf{State} 0 - limit line disabled
  1 - limit line enabled.
  - \textbf{Fixed/Rel} 0 - Relative
  1 - Fixed
  - \textbf{Upper/Lower} 0 - Upper limit
  1 - Lower limit
  - $nP =$ Number of points to follow
  - \textbf{Freq value} X-axis value
  - \textbf{Amp value} Y-axis value

Subsequent points are appended to the data in the same manner.

**Examples**

- The following writes one relative, upper limit segment with 4 points for a 40 MHz bandpass filter.
- Blue-shaded numbers are values for one segment.
  Following are X/Y points
  \texttt{CALC:LIM:LLD 1,1,0,0,4,-30e6,-20,-20e6,-10,-10e6,0,10e6,0,20e6,-10,30e6,-20}

**Query Syntax** \texttt{CALCulate[:SELection]:LIMit:LLData?}

**Return Type** Block data

**Default** 0 - Limit line data off

---

**Last Modified:**

01June2018 Added NF mode Opt. 356 (10.3)

17-Jul-2012 New command
**CALCulate[:SESelecte]d:]LIIMit:SOUNd <char>**

**Note:** This command replaces `CALCulate[:SESelecte]d:]LIIMit:BEERP.

(Read-Write) Set and query the conditions with which the FieldFox beep occurs during limit line testing. For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SESelect`.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>CAT, NA, NF, SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>&lt;char&gt; Beep state. Choose from:</td>
<td></td>
</tr>
<tr>
<td>OFF - No beeping</td>
<td></td>
</tr>
<tr>
<td>OPASs - Beep on Pass</td>
<td></td>
</tr>
<tr>
<td>OFAiil - Beep on Fail</td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td><code>CALC:LIM:SOUN OPA</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td><code>CALCulate:SESelected:LIIMit:SOUNd?</code></td>
</tr>
<tr>
<td>Return Type</td>
<td>Character</td>
</tr>
<tr>
<td>Default</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Last Modified:

- 01June2018 Added NF mode Opt. 356 (10.3)
- 16-Apr-2013 New command (6.25)

**CALCulate[:SESelecte]d:]LIIMit[:STATe]**

(Read-Write) Set and query whether limit testing occurs. For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SESelect`.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>CAT, NA, NF, SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>&lt;bool&gt; Testing state. Choose from:</td>
<td></td>
</tr>
<tr>
<td>0 or OFF - No limit line testing</td>
<td></td>
</tr>
<tr>
<td>1 or ON - Do limit line testing</td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td><code>CALC:LIM ON</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td><code>CALCulate[:SESelecte]d:]LIIMit[:STATe]?</code></td>
</tr>
<tr>
<td>Return Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Last Modified:
CALCulate[:SELected]:LIMit:WARN <bool>

*(Read-Write)* Set and query whether the Pass and Fail warning is displayed on the FieldFox screen. For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELect.

**Relevant Modes** CAT, NA, NF, SA

**Parameters**

<bool> Testing state. Choose from:

0 or **OFF** - Do NOT display onscreen warning.

1 or **ON** - Display onscreen warning.

**Examples** CALC:LIM:WARN ON

**Query Syntax** CALCulate[:SELected]:LIMit:WARN?

**Return Type** Boolean

**Default** OFF

---

01june2018 Added NF mode Opt. 356 (10.3)

CALCulate[:SELected]:MARKer<n>:ACTivate

*(Write-Only)* Makes the selected marker active. For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELect.

**Relevant Modes** CAT, NA, NF, SA, RTSA *Pulse Measurements, FOPS*  
**Note:** SA & RTSA modes do NOT recognize the optional [:SELected] node.

**Parameter**

<n> Marker number to activate. Each trace can contain up to six markers.

**Examples** CALC:MARK2:ACTivate

**Query Syntax** Not Applicable

**Default** Not Applicable

---

01june2018 Added NF mode Opt. 356 (10.3)
### CALCulate[:SELection]:MARKer:AOFF

(Write-Only) Turns OFF all markers.

**Relevant Modes**

- CAT, NA, **NF**, SA, RTSA  
  **Note:** SA & RTSA modes do NOT recognize the optional [:SELection] node.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:MARK:AOFF</td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

**Last Modified:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-june-2018</td>
<td>Added Opt. 356 NF(10.3)</td>
</tr>
<tr>
<td>10-june-2016</td>
<td>Added RTSA mode (9.50)</td>
</tr>
</tbody>
</table>

---

### CALCulate[:SELection]:MARKer:BWIDth:DATA?

(Read-only) Read the results of the bandwidth marker search. Returns Bandwidth, Center Frequency, Q, and Loss. Use **CALCulate:MARKer:BWIDth[:STATe]** to create bandwidth markers.

This command operates on the selected trace. First select a trace using **CALC:PAR<tr>:SELection**.

**Relevant Modes**

- NA

**Parameters**

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:MARK:BWID:DATA?</td>
<td></td>
</tr>
</tbody>
</table>

**Return Type**

4 Numeric values separated by commas.

**Default**

Not Applicable

---

**Last Modified:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13-Aug-2012</td>
<td>Added select</td>
</tr>
<tr>
<td>2-Aug-2011</td>
<td>New command A.05.50</td>
</tr>
</tbody>
</table>
CALCulate[:SELECTed]:MARKer:BWIdth[:STATe] <bool>

(Read-Write) Set or return the state of a bandwidth marker search. This command operates on the selected trace. First select a trace using CALCulate:PARmeter<tr>:SELECT.

Use CALCulate:MARKer:FUNCTION:BWIdth:THReshold to set the bandpass or notch value to be used in the search.

Use CALCulate:MARKer:BWIdth:DATA to read the data.

Relevant Modes NA

Parameters

<bool> Choose from:

ON (or 1) - Uses markers 1 through 4 to perform a bandwidth search.
OFF (or 0) - Turns OFF the BW search. Use CALCulate:MARKer:AOFF to turn the markers OFF.

Examples CALC:MARK:BWID ON

Query Syntax CALCulate[:SELECTed]:MARKer:BWIdth[:STATe]?

Return Type Boolean

Default Off

Last Modified:

13-Aug-2012 Added select
1-Aug-2011 New command (A.05.50)

CALCulate[:SELECTed]:MARKer:COUPled <bool>

(Read-Write) Set and query the state of marker coupling. Marker coupling is used in multi-trace configurations. This setting affects all NA mode markers.

Relevant Modes NA, NF

Parameters

<bool> Choose from:

OFF or 0 - Markers are uncoupled.
ON or 1 - Markers are coupled.

Examples CALC:MARK:COUP OFF

Query Syntax CALCulate[:SELECTed]:MARKer:COUPled?

Return Type Boolean

Default ON

Last Modified:
CALCulate[:SELection]:MARKer:FORMat <char>

(Read-Write) Set and query marker format that appears in the upper-right of the screen and in the marker table. This format can be different from the displayed format. All of the markers on the selected trace change to the specified format. Use CALC:MARK:Y? to read the Y-axis values that are on the screen.

Relevant Modes: NA

Parameters

<char> Marker format. Choose from:
  DEF - (Default) Same as displayed format.
  IMPedance - R+jX format
  PHASE - Phase in degrees.
  ZMAGnitude - Impedance Magnitude
  MAGPhase - Magnitude and Phase
  REAL -
  IMAGinary -
  DBA -

Examples

CALC:MARK:FORM IMP

Query Syntax

CALCulate[:SELection]:MARKer:FORMat?

Return Type

Character

Default

Depends on model and measurement.

Last Modified:

4-Jun-2013 New topic (A.07.00)

CALCulate[:SELection]:MARKer:FUNCtion:BWIDth:THReshold <value>

(Read-Write) Set or return the value used to find the bandwidth of a bandpass or notch filter response. Use CALCulate:MARKer:BWIDth[:STATe] to turn BW search On/Off. Use CALCulate:MARKer:BWIDth:DATA to read the data.

Relevant Modes: NA

Parameters

<value> BW search value. Specify the level in dB from the peak or valley where
bandwidth is measured. Negative numbers search for a Peak bandpass, such as a filter S21 response. Either of the following TWO methods can be used to search for a Valley or Notch filter, such as the S11 response of a bandpass filter.

- Negative values search down from the TOP (MAX) of the response.
- Positive values search up from the BOTTOM (MIN) of the notch.

**Examples**  
`CALC:MARK1:FUNC:BWID:THR 3`

**Query Syntax**  
`CALCulate[:SELected]:MARKer:FUNCtion:BWIDth:THReshold?`

**Return Type** Numeric

**Default** -3

---

**CALCulate[:SELeected]:MARKer<n>:FUNCtion:FALLtime <num>**

*(Write-Read)* Set and query pulse drop in dBm. Marker 1 is created as a delta marker and placed at the first instance of the peak value and the specified fall on the trace. Read the fall time using `CALCulate[:SELeected]:MARKer<X>?`

**Relevant Modes** Pulse Measurements

**Parameters**

- `<num>` Pulse Fall in dBm.

**Examples**  
`CALC:MARK:FUNC:FALL -5`

**Query Syntax**  
`CALCulate[:SELeected]:MARKer<n>:FUNCtion:FALLtime?`

**Default** -3 dBm

---

**CALCulate[:SELeected]:MARKer<n>:FUNCtion:MAXimum**

*(Write-Only)* Causes the specified marker to find the highest (maximum) amplitude of the trace.

**Relevant Modes** CAT, NA, NF, SA, RTSA, Pulse Measurements, FOPS

**Note:** SA & RTSA modes do not recognize the optional [:SELeected] node.

**Parameters**
### CALCulate[:SELECTed]:MARKer<n>:FUNCTION:MINimum

*Write-Only* Causes the specified marker to find the lowest (minimum) amplitude of the trace. For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELECT`.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>CAT, NA, NF, SA, RTSA, Pulse Measurements, FOPS</th>
</tr>
</thead>
</table>

**Note:** SA & RTSA modes do NOT recognize the optional [:SELECTed] node.

<table>
<thead>
<tr>
<th>Parameters</th>
<th><code>&lt;n&gt;</code> New or existing marker to move. Choose from 1 to 6.</th>
</tr>
</thead>
</table>

#### Examples

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CALC:MARK1:FUNC:MAX</code></td>
</tr>
</tbody>
</table>

#### Query Syntax

Not Applicable

#### Default

Not Applicable

---

**Last Modified:**

- 01 June 2018: Added NF mode Opt. 356 (10.3)
- 10 June 2016: Added RTSA mode (9.50)
- 21 March 2014: Added FOPS
- 29 October 2013: Added Pulse

---

### CALCulate[:SELECTed]:MARKer:FUNCTION:PEXCursion <num>

*Read-Write* Set and query the excursion value. This is the vertical distance (dB) between the peak and the valleys on both sides. To be considered a peak for subsequent marker search commands, data values must "fall off" from the peak on both sides by the excursion value.

For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELECT`.

#### Parameters

<table>
<thead>
<tr>
<th><code>&lt;n&gt;</code></th>
<th>New or existing marker to move. Choose from 1 to 6.</th>
</tr>
</thead>
</table>

#### Examples

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CALC:MARK1:FUNC:MIN</code></td>
</tr>
</tbody>
</table>

#### Query Syntax

Not Applicable

#### Default

Not Applicable

---

**Last Modified:**

- 01 June 2018: Added NF mode Opt. 356 (10.3)
- 10 June 2016: Added RTSA mode (9.50)
- 21 March 2014: Added FOPS
- 29 October 2013: Added Pulse

---
**CALC:PAR<tr>:SELect**.

Peak Excursion and Threshold settings apply to all markers on each trace. These settings remain through an instrument preset, but are reset to the default values when the FieldFox power is shutdown.

**Relevant Modes**  
CAT, NA, NF, SA, RTSA, Pulse Measurements  
*Note: SA & RTSA modes do NOT recognize the optional [:SElected] node.*

**Parameters**

<num>  
Excursion value in dB.

**Examples**  
CALC:MARK:FUNC:PEXC 3

**Query Syntax**  
CALCulate[:SELected]:MARKer:FUNCtion:PEXCursion?

**Return Type**  
Numeric

**Default**  
0.5 dB

---

Last Modified:

01june2018  
Added NF mode Opt. 356 (10.3)

10-june-2016  
Added RTSA mode (9.50)

29-Oct-2013  
Added Pulse

---

**CALCulate[:SELected]:MARKer<n>:FUNCtion:PLEFt**  
*(Write-Only)* Causes the specified marker to find the next data point to the left that meets the 'Peak' criteria. When no data points to the left meet the Peak criteria, the marker does not move.

**Relevant Modes**  
NF, SA, RTSA  
*Note: SA & RTSA modes do NOT recognize the optional [:SElected] node.*

**Parameters**

<n>  
New or existing marker to move. Choose from 1 to 6.

**Examples**  
CALC:MARK1:FUNC:PLEFT

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

Last Modified:

01june2018  
Added NF mode Opt. 356 (10.3)

10-june-2016  
Added RTSA mode (9.50)
(Write-Only) Causes the specified marker to find the next lower amplitude peak from where it currently resides. This command is usually preceded with the \texttt{MARK:FUNC:MAX} command which finds the highest peak.

For NA and CAT mode, this command operates on the selected trace. First select a trace using \texttt{CALC:PAR<tr>:SELect}.

**Relevant Modes** CAT, NA, NF, SA, RTSA \textbf{Pulse Measurements}  
\textbf{Note:} SA & RTSA modes do NOT recognize the optional [:SELected] node.

**Parameters**
\texttt{<n>} New or existing marker to move. Choose from 1 to 6.

**Examples** \texttt{CALC:MARK1:FUNC:PNEX}

**Query Syntax** Not Applicable  
**Default** Not Applicable

**Last Modified:**
01june2018 Added NF mode Opt. 356 (10.3)  
10-june-2016 Added RTSA mode (9.50)  
29-Oct-2013 Added Pulse

\texttt{CALCulate[:SELected]:MARKer<n>:FUNCtion:PRIGht}

(Write-Only) Causes the specified marker to search to the right of the current location for the next data point that meets the ‘Peak’ criteria.

**Relevant Modes** NF, SA, RTSA

**Parameters**
\texttt{<n>} New or existing marker to move. Choose from 1 to 6.

**Examples** \texttt{CALC:MARK1:FUNC:PRIG}

**Query Syntax** Not Applicable  
**Default** Not Applicable

**Last Modified:**
01june2018 Added NF mode Opt. 356 (10.3)  
10-june-2016 Added RTSA mode (9.50)

\texttt{CALCulate[:SELected]:MARKer:FUNCtion:PTHReshold <num>}

(Read-Write) The minimum amplitude used to find peaks in subsequent marker search commands. To be considered valid, the peak must be \textbf{above} the threshold level. The valley on either side can be below the threshold level.
For NA, NF, and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELect`.

Peak Excursion and Threshold settings apply to all markers on each trace. These settings remain through an instrument preset, but are reset to the default values when the FieldFox power is shutdown.

**Relevant Modes**  CAT, NA, NF, SA, RTSA **Pulse Measurements**

**Note:** SA & RTSA modes do NOT recognize the optional [:SELected] node.

**Parameters**

- `<num>`: Threshold value in dB.

**Examples**

```
CALC:MARK:FUNC:PTH  -10
```

**Query Syntax**

```
CALCulate:SELected:MARKer:FUNCTION:PTHreshold?
```

**Default**

-190 dB

---

**Last Modified:**

- 01 June 2018: Added NF mode Opt. 356 (10.3)
- 10 June 2016: Added RTSA mode (9.50)
- 29 Oct 2013: Added Pulse

---

**CALCulate[:SELected]:MARKer:FUNCTION:RISetime <num>**

**(Write-Read)** Set and query pulse rise in dBm. Marker 1 is created as a delta marker and placed at the first instance of the peak value and the specified rise on the trace. Read the rise time using `CALCulate[:SELected]:MARKer<X`?

**Relevant Modes**  **Pulse Measurements**

**Parameters**

- `<num>`: Pulse Rise in dBm.

**Examples**

```
CALC:MARK:FUNC:RIS  5
```

**Query Syntax**

```
CALCulate[:SELected]:MARKer<n>:FUNCTION:RISetime?
```

**Default**

- dBm

---

**Last Modified:**

- 29 Oct 2013: Added Pulse
(Read-Write) Set and query the marker target search value.
This command operates on the selected trace AND the active marker.
First select a trace using `CALC:PAR<tr>:SELect`.
Then activate a marker using `CALCulate[:SELected]:MARKer:ACTivate`.

**Relevant Modes**  NA, NF, Pulse Measurements

**Parameters**

*<value>*  Target search value in dB.

**Examples**  `CALC:MARK:FUNC:TARG -1`

**Query Syntax**  `CALCulate[:SELected]:MARKer:FUNCtion:TARGet?`

**Return Type**  Numeric  
**Default**  -3 dB

---

**CALCulate[:SELected]:MARKer:FUNCtion:TDIRection <value>**

(Read-Write) Set and query whether the target search will 'wrap' when it gets to the end of the range.
This command operates on the selected trace, AND the active marker, AND only a 'Target' search.

**Note:** There is NO front-panel GUI equivalent for this command.

1. Select a trace using `CALC:PAR<tr>:SELect`.
2. Then activate a marker using `CALCulate[:SELected]:MARKer:ACTivate`.
3. Then start a search using `CALCulate[:SELected]:MARKer:FUNCtion:TARGet`.

**Relevant Modes**  NA, Pulse Measurements

**Parameters**

*<value>*  Choose from:

**DEFAULT** - Search from the current marker position to the right. Wrap around to the beginning and continue search.
**LEFT** - Search to the left from the current position. When the range start is reached, end the search (no wrap).

**RIGHT** - Search to the right from the current position. When the range stop is reached, end the search (no wrap).

**Examples**

```
CALC:MARK:FUNC:TDIR LEFT
```

**Query Syntax**

CALCulate[:SELected]:MARKer:FUNCtion:TDIRection?

**Return Type**

Character

**Default**

Default

---

**CALCulate[:SELelected]:MARKer<n>:FUNCtion:TRACking <bool>**

*(Read-Write)* Set and query the marker tracking ON/OFF state. Marker tracking can be used with all search functions.

For NA mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELect`.

**Relevant Modes**

NA, NF, Pulse Measurements

**Parameters**

- `<n>` Marker number for which tracking is to be set. A marker is created if not already ON. Choose from 1 to 6. If unspecified, value is set to 1.
- `<bool>` Marker tracking state. Choose from:
  - **ON** (or 1) - Marker Tracking ON.
  - **OFF** (or 0) - Marker Tracking OFF.

**Examples**

```
CALC:MARK2:FUNC:TRAC 1
```

**Query Syntax**

CALCulate:SELelected:MARKer<n>:FUNCtion:TRACKing?

**Return Type**

Boolean

**Default**

OFF

---

**Last Modified:**

- 01-june-2018 Added Opt. 356 NF(10.3)
- 29-Oct-2013 Added Pulse
- 2-Aug-2011 New command (5.50)
CALCulate[:SELeected]:MARKer:FUNCtion:ZONE <bool>

(Write-Read) Set and query whether marker function searches occur on the entire primary trace graph or within the zoom window time span.

Relevant Modes  Pulse Measurements

Parameters

<bool>  Choose from:

OFF or 0 - Marker function searches occur on the entire primary trace graph
ON or 1 - Marker function searches occur within the zoom window time span.

Examples   CALC:MARK:FUNC:ZONE 1

Query Syntax  CALCulate[:SELeected]:MARKer:FUNCtion:ZONE?

Default    OFF

---

CALCulate[:SELeected]:MARKer<n>:SET <char>

(Write-Only) Sets the FieldFox setting <char> to the location of the specified marker.
For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELect.

Note: SA mode does NOT recognize the optional [:SELeected] node.

Relevant Modes  The following <char> arguments are valid in each mode:

An error is returned if <char> is not valid.

<table>
<thead>
<tr>
<th>Mode/function</th>
<th>Valid &lt;char&gt; Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT (non-DTF)</td>
<td>START, STOP, CENTer, RLEVel</td>
</tr>
<tr>
<td>CAT (DTF)</td>
<td>START, STOP, RLEVel</td>
</tr>
<tr>
<td>NA</td>
<td>START, STOP, CENTer, DEL</td>
</tr>
<tr>
<td><strong>NF</strong></td>
<td>START, STOP, CENTer, DEL, RLEVel</td>
</tr>
<tr>
<td>SA (Non-Zerospan)</td>
<td>START, STOP, CENTer, RLEVel</td>
</tr>
<tr>
<td>SA (Zerospan)</td>
<td>RLEVel</td>
</tr>
<tr>
<td>SA (Tune and Listen)</td>
<td>TLFRequency</td>
</tr>
</tbody>
</table>

---

Parameters
<n> Existing marker location from which <char> will be set. Choose from 1 to 6.

<char>

START - Sets the sweep Start value to the marker X-axis value.
STOP - Sets the sweep Stop value to the marker X-axis value.
CENTer - Sets the center frequency of the sweep to the marker X-axis value.
DELay - Sets the electrical delay to that of the current marker group delay value.
RLEVEL - Sets the reference level value to the marker Y-axis value.
TLFRequency - Sets the SA Mode Tune and Listen frequency to the marker X-axis value.

Examples

CALC:MARK1:SET CENT

Query Syntax

Not Applicable

Default

Not Applicable

Last Modified:

01 June 2018 Added NF mode Opt. 356 (10.3)
19 Oct 2010 Updated with DEL (5.30)

CALCulate[:SELeected]:MARKer<n>:SET:CENTer

(Write-Only) The center frequency of the measurement becomes the value of the specified marker. The frequency span is adjusted accordingly.

CALCulate[:SELeected]:MARKer:SET performs the same operation.

Relevant Modes NF, SA, RTSA

Parameters

<n> Existing marker from which the center frequency will be set. Choose from 1 to 6.

Examples

CALC:MARK1:SET CENT

Query Syntax

Not Applicable

Default

Not Applicable

Last Modified:

01 June 2018 Added NF mode Opt. 356 (10.3)
10 June 2016 Added RTSA mode (9.50)

CALCulate[:SELeected]:MARKer<n>:SET:REFLevel

(Write-Only) The reference level of the current window becomes the value of the specified marker.
CALCulate[:SELection]:MARKer:SET performs the same operation.

**Relevant Modes**  NF, SA, RTSA

**Parameters**

<n> Existing marker for which reference level will be set. Choose from 1 to 6.

**Examples**  CALC:MARK1:SET:REFL

**Query Syntax**  Not Applicable

**Default**  Not Applicable

---

Last Modified:

- 01june2018  Added NF mode Opt. 356 (10.3)
- 10-june-2016  Added RTSA mode (9.50)

---

CALCulate[:SELection]:MARKer<n>[:STATe] <char>

(Read-Write) Create, change, or remove a marker.

For NA and CAT mode, this command operates on the selected trace. First select a trace using CALC:PAR<tr>:SELect.

**Relevant Modes**  CAT, NA, NF, SA, RTSA, Pulse Measurements, FOPS

| Note: SA, RTSA modes do NOT recognize the optional [:SELection] node. |

**Parameters**

<n> New or existing marker to create, change, or remove. Choose from 1 to 6.

<char> Choose from:

- NORM - Marker is a Normal marker
- DELT - Marker is a Delta marker pair.
- OFF - Marker is disabled.

**To create and move a delta marker:**

1. Create a normal marker using this command.
2. Move the marker to the reference position using CALCulate:MARKer:X
3. Change the marker to a delta marker using this command.
4. Move the delta marker to the delta position using CALCulate:MARKer:X

**Examples**  CALC:MARK1 NORM

**Query Syntax**  CALCulate[:SELection]:MARKer<n>[:STATe]?

**Return Type**  Character

**Default**  Off
CALCulate[:SELEc]:MARKer<n>:TRACe <tnum>

*(Read-Write)* Moves an existing marker to the specified trace number. The marker and the CAT or NA trace must exist already. This feature is called Marker Trace in the User's Guide.

For NA and CAT mode, this command moves the selected trace. First select a trace using `CALC:PAR<tr>:SELEc`.

**Relevant Modes**  CAT, NA, NF, SA, RTSA

Note: SA & RTSA modes do NOT recognize the optional [:SELEc] node.

**Parameters**

- `<n>` Existing marker to assign to trace. Choose from 1 to 6.
- `<tnum>` Trace number. Choose from:
  - CAT, NA, and NF:
    - 0 - Auto. Marker resides on data trace when it is visible. Marker resides on memory trace when only IT is visible.
    - 1 - Data Trace
    - 2 - Memory Trace
  - SA:
    - 1, 2, 3, 4

**Examples**

```
CALC:MARK1:TRAC 1
```

**Query Syntax**

`CALCulate[:SELEc]:MARKer<n>:TRACe?`

**Default**

1

---

Last Modified:

01 June 2018   Added NF mode Opt. 356 (10.3)
10 June 2016   Added RTSA mode (9.50)
20 Mar 2014    Added FOPS
29 Oct 2013    Added Pulse

---

CALCulate[:SELEc]:MARKer<n>:X <num>

*(Read-Write)* Set and query the X-axis location for the specified marker. See To Create and Move a Delta Marker

For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELEc`.

---

Last Modified:

01 June 2018   Added NF mode Opt. 356 (10.3)
10 June 2016   Added RTSA mode (9.50)
### Relevant Modes

CAT, NA, **NF**, SA, RTSA **Pulse Measurements**, FOPS

**Note:** SA, RTSA modes do NOT recognize the optional [:SELelected] node.

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;n&gt;</td>
<td>Existing marker for which to set X-axis location. Choose from 1 to 6.</td>
</tr>
<tr>
<td>&lt;num&gt;</td>
<td>X-axis location. Choose any value currently displayed on the X-axis.</td>
</tr>
</tbody>
</table>

**Note:** In CAT or NA mode, units of **time** (Freq span of zero Hz.) are **NOT** accepted.

### Examples

```
CALC:MARK1:X 4e9
```

### Query Syntax

`CALCulate[:SELection]:MARKer<n>:X?`

### Return Type

Numeric

### Default

When created, markers appear at the center of the X-axis.

---

**Last Modified:**

- 01June2018: Added NF mode Opt. 356 (10.3)
- 10-June-2016: Added RTSA mode (9.50)
- 20-Mar-2014: Added FOPS

### CALCulate[:SELect]:MARKer<n>:Y?

*(Read-Only)* Reads the Y-axis value for the specified marker.

For NA and CAT mode, this command operates on the selected trace. First select a trace using `CALCulate[:SELection]:PAR<tr>:SELection`.

**Note:** SA & RTSA modes do NOT recognize the optional [:SELection] node.

#### Relevant Modes

FOPS, CAT, **NF** - Two values are returned: Magnitude and zero.

NA - Format depends on `CALCulate[:SELection]:MARKer:FORMat`

SA & RTSA - One value is returned: Magnitude

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;n&gt;</td>
<td>Existing marker for which to read Y-axis value. Choose from 1 to 6.</td>
</tr>
</tbody>
</table>

#### Examples

```
CALC:MARK1:Y?
```

#### Return Type

Numeric

#### Default

Not Applicable

---

**Last Modified:**

- 01June2018: Added NF mode Opt. 356 (10.3)
- 10-June-2016: Added RTSA mode (9.50)
- 20-Mar-2014: Added FOPS
**CALCulate[:SELected]:MARKer:Y<n>:UNCertainty?**

*(Read-Only)* Reads the Y-axis uncertainty value for the specified marker. This command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELect`.

**Relevant Modes**  
NF - One value is returned for symmetric values (Noise Factor and Noise Temperature) and two values are returned for asymmetric values (Noise Figure).

**Parameters**

<n>  
Existing marker for which to read Y-axis value. Choose from 1 to 2.

**Examples**  
`CALC:MARK1:Y:UNC?`

**Return Type**  
Numeric

**Default**  
Not Applicable

---

Last Modified:  
01 June 2018  
Added NF mode Opt. 356 (10.3)

**CALCulate[:SELected]:MATH:FUNCtion <char>**

*(Read-Write)* Set and query the math function. A trace must already be stored into memory (CALC:MATH:MEM) for a setting other than NORM. This command operates on the selected trace. First select a trace using `CALC:PAR<tr>:SELect`.

**Relevant Modes**  
NA, CAT

**Parameters**

<char>  
Choose from:  
NORM - Math Off  
ADD - (Data + Memory)  
SUBTract - (Data - Memory)  
DIVide - (Data / Memory)

**Examples**  
`CALC:MATH:FUNC DIV`

**Query Syntax**  
`CALCulate[:SELected]:MATH:FUNCtion?`

**Return Type**  
Character

**Default**  
NORM

---

**CALCulate[:SELected]:SMOothing:APERture <num>**
(Read-Write) Set and query the smoothing aperture for the selected trace.
This command operates on the selected trace. First select a trace using \texttt{CALC:PAR<tr>:SESelect}. Use \texttt{CALCulate[:SESelect]:SMOoothing[:STATe]} to enable and disable smoothing.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NA, CAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;num&gt; Trace smoothing in percent. Choose a value between 0 and 25.</td>
</tr>
<tr>
<td></td>
<td>Examples</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>\texttt{CALCulate[:SESelect]:SMOoothing:APERture?}</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Last Modified:
20-Jan-2015 Added CAT
19-Oct-2010 New command (5.30)

\texttt{CALCulate[:SESelect]:SMOoothing[:STATe] <bool>}

(Read-Write) Set and query whether trace smoothing occurs on the selected trace. Not available with Polar or Smith Chart formats.
This command operates on the selected trace. First select a trace using \texttt{CALC:PAR<tr>:SESelect}. Use \texttt{CALCulate[:SESelect]:SMOoothing:APERture} to set aperture.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NA, CAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;bool&gt; Testing state. Choose from: 0 or \texttt{OFF} - Smoothing disabled. 1 or \texttt{ON} - Smoothing enabled.</td>
</tr>
<tr>
<td></td>
<td>Examples</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>\texttt{CALCulate[:SESelect]:SMOoothing[:STATe]?}</td>
</tr>
<tr>
<td>Return Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Last Modified:
20-Jan-2015 Added CAT
19-Oct-2010 New command (5.30)
**CALCulate[:SELected]:TIME:AUX:CENTer <num>**

(Read-Write) Set and query the center time of the zoom window.

**Relevant Modes**  
Pulse Measurements

**Parameters**

<num>  Center time in seconds.

**Examples**  
CALC:TIME:AUX:CENT 5e-5

**Query Syntax**  
CALCulate[:SELected]:TIME:AUX:CENTer?

**Return Type**  
Numeric

**Default**  
5e-5

---

Last Modified:
29-Oct-2013  New command

**CALCulate[:SELected]:TIME:AUX:PDIVision <num>**

(Read-Write) Set and query the Time per division of the zoom window.

**Relevant Modes**  
Pulse Measurements

**Parameters**

<num>  Per division time in seconds.

**Examples**  
CALC:TIME:AUX:PDIV 10e-6

**Query Syntax**  
CALCulate[:SELected]:TIME:AUX:CENTer?

**Return Type**  
Numeric

**Default**  
5e-5

---

Last Modified:
29-Oct-2013  New command

**CALCulate[:SELected]:TIME:CENTer <num>**

(Read-Write) Set and query the center time of the primary trace graph window.

**Relevant Modes**  
Pulse Measurements

**Parameters**

<num>  Center time in seconds.
### CALCulate[:SELected]:TIME:CENTer

<table>
<thead>
<tr>
<th>Examples</th>
<th>CALC:TIME:CENT 5e-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>CALCulate[:SELected]:TIME:CENTer?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>5e-5</td>
</tr>
</tbody>
</table>

Last Modified:

29-Oct-2013 New command

### CALCulate[:SELected]:TIME:LENGth `<num>`

(Read-Write) Set and query the time span of the primary trace graph window.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Pulse Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td><code>&lt;num&gt;</code> Time span in seconds.</td>
</tr>
<tr>
<td>Examples</td>
<td>CALC:TIME:LENG 5e-5</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>CALCulate[:SELected]:TIME:LENGth?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>100e-6</td>
</tr>
</tbody>
</table>

Last Modified:

29-Oct-2013 New command

### CALCulate[:SELected]:TIME:PDIVision `<num>`

(Read-Write) Set and query the Time per division of the primary trace graph window.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Pulse Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td><code>&lt;num&gt;</code> Per division time in seconds.</td>
</tr>
<tr>
<td>Examples</td>
<td>CALC:TIME:PDIV 1e-6</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>CALCulate[:SELected]:TIME:PDIVision?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>10e-6</td>
</tr>
</tbody>
</table>
**CALCulate[:SELected]:TIME:STARt <num>**

*(Read-Write)* Set and query the start time of the primary trace graph window.

**Relevant Modes**  Pulse Measurements

**Parameters**

- `<num>`: Start time in seconds.

**Examples**

```
CALC:TIME:STAR 1e-9
```

**Query Syntax**

```
CALCulate[:SELected]:TIME:STARt?
```

**Return Type**  Numeric

**Default**  0

---

**CALCulate[:SELected]:TRACe<n>:DATA?**

*(Read-only)* Read the data from a Trace Graph measurement.

**Relevant Modes**  Pulse Measurements, FOPS

**Parameters**

- `<n>`: Trace number.
  - For Pulse, choose from the following:
    - 1 - Primary trace graph
    - 2 - Zoom window
  - For FOPS, choose 1 or unspecify.

**Examples**

```
CALC:TRAC:DATA?
```

**Query Syntax**

```
CALCulate[:SELected]:TRACe:DATA?
```

**Return Type**  Numeric

**Default**  Not Applicable

---

Last Modified:

- 29-Oct-2013  New command
- 21-Mar-2014  Added FOPS
- 29-Oct-2013  New command
**CALCulate[:SELeCted]:TRANsform:DISTance:BANDpass <bool>**

*(Read-Write)* Set and query Bandpass mode for Distance to Fault measurements. This command performs the same function as `CALCulate:TRANsform:FREQuency[:TYPE]`.

**Relevant Modes**  
CAT

**Parameters**  

- `<bool>`  
  Bandpass mode. Choose from:  
  - 0 or OFF - Lowpass mode  
  - 1 or ON - Bandpass mode

**Examples**  

CALC:TRAN:DIST:BAND 1

**Query Syntax**  
CALCulate[:SELeCted]:TRANsform:DISTance:BANDpass?

**Return Type**  
Boolean

**Default**  
1 - ON

---

**CALCulate[:SELeCted]:TRANsform:DISTance:FREQuency:CENTer <num>**

*(Read-Write)* Set and query the bandpass center frequency for Distance to Fault measurements. Set Bandpass mode using **CALC:TRAN:DIST:BANDpass ON**.

**Relevant Modes**  
CAT

**Parameters**  

- `<num>`  
  Bandpass center frequency in Hz.

**Examples**  

CALC:TRAN:DIST:FREQ:CENT 1e9

**Query Syntax**  
CALCulate[:SELeCted]:TRANsform:DISTance:FREQuency:CENTer?

**Return Type**  
Numeric

**Default**  
Dependent on DTF distance.

---

**CALCulate[:SELeCted]:TRANsform:DISTance:FREQuency:SPAN:MAXimum <num>**

*(Read-Write)* Set and query the bandpass maximum frequency span for Distance to Fault measurements. Set Bandpass mode using **CALC:TRAN:DIST:BANDpass ON**.

**Relevant Modes**  
CAT

**Parameters**  

- `<num>`  
  Bandpass maximum frequency span in Hz.

**Examples**  

CALC:TRAN:DIST:FREQ:SPAN:MAX 100e6

**Query Syntax**  
CALCulate[:SELeCted]:TRANsform:DISTance:FREQuency:SPAN:MAXimum?

**Return Type**  
Numeric

**Default**  
Dependent on DTF distance.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Relevant Modes</th>
<th>Parameters</th>
<th>Examples</th>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCulate[:SELelected]:TRANSform:DISTance:FREQuency:STARt:MINimum &lt;num&gt;</td>
<td>(Read-Write) Set and query the start frequency value in Bandpass mode for a Distance to Fault measurement.</td>
<td>CAT</td>
<td>&lt;num&gt; Start value in hertz. This command will accept MIN and MAX as arguments.</td>
<td>CALC:TRAN:DIST:FREQ:STAR:MIN 1GHz</td>
<td>CALCulate[:SELelected]:TRANSform:DISTance:FREQuency:STARt:MINimum?</td>
<td>Numeric</td>
<td>2 MHz</td>
</tr>
<tr>
<td>CALCulate[:SELelected]:TRANSform:DISTance:FREQuency:STOP:MAXimum &lt;num&gt;</td>
<td>(Read-Write) Set and query the stop frequency value in Bandpass mode for a Distance to Fault measurement.</td>
<td>CAT</td>
<td>&lt;num&gt; Stop value in hertz. This command will accept MIN and MAX as arguments.</td>
<td>CALC:TRAN:DIST:FREQ:STOP:MAX 2GHz</td>
<td>CALCulate[:SELelected]:TRANSform:DISTance:FREQuency:STOP:MAXimum?</td>
<td>Numeric</td>
<td>FieldFox maximum frequency</td>
</tr>
<tr>
<td>CALCulate[:SELelected]:TRANSform:DISTance:STARt &lt;num&gt;</td>
<td>(Read-Write) Set and query the X-axis start value in a Distance to Fault measurement.</td>
<td>CAT</td>
<td>&lt;num&gt; Start value in meters. Choose a value between the MIN distance and the Stop distance. (This command will accept MIN and MAX as arguments.)</td>
<td>CALC:TRAN:DIST:STAR 10</td>
<td>CALCulate[:SELelected]:TRANSform:DISTance:STARt?</td>
<td>Numeric</td>
<td>0</td>
</tr>
</tbody>
</table>
**CALCulate[:SELected]:TRANSform:DISTance:STOP <num>**

*(Read-Write)* Set and query the X-axis stop value in Distance to Fault measurement.

**Relevant Modes** CAT

**Parameters**

- **<num>** Stop value in meters. Choose a value between the Start distance and the MAX distance. (This command will accept MIN and MAX as arguments.)

**Examples**

```
CALC:TRAN:DIST:STOP .20
```

**Query Syntax**

```
CALCulate[:SELected]:TRANSform:DISTance:STOP?
```

This value will always be returned in meters - never in feet.

**Return Type** Numeric

**Default** 100 meters

---

**CALCulate[:SELected]:TRANSform:DISTance:UNIT <char>**

*(Read-Write)* Set and query the X-axis units in Distance to Fault measurement.

**Note:** DTF **Start** and **Stop** settings always return distance in meters.

**Relevant Modes** CAT

**Parameters**

- **<char>** X-axis units. Choose from:
  - METers
  - FEET

**Examples**

```
CALC:TRAN:DIST:UNIT
```

**Query Syntax**

```
CALCulate[:SELected]:TRANSform:DISTance:UNIT?
```

**Return Type** Character

**Default** METers

---

Last Modified:

14-Feb-2013 Added note

---

**CALCulate[:SELected]:TRANSform:DISTance:WINDow <char>**

*(Read-Write)* Set and query the DTF window setting.

**Relevant Modes** CAT

**Parameters**
DTF window setting. Choose from:
- **RECT** - Minimum window setting.
- **HAMM** - Medium window setting.
- **KBES** - Maximum window setting.

**Examples**
```
CALC:TRAN:DIST:WIND KBES
```

**Query Syntax**
```
CALCulate[:SELected]:TRANsform:DISTance:WINDow?
```

**Return Type**
Character

**Default**
KBES

---

**CALCulate[:SELected]:TRANsform:FREQuency[:TYPE] <char>**

*(Read-Write)* Set and query the DTF Frequency mode. This command performs the same function as `CALCulate:TRANsform:DISTance:BANDPass`.

**Relevant Modes**
CAT

**Parameters**
- `<char>`
  - DTF frequency mode. Choose from:
    - **BPASs** - Bandpass mode.
    - **LPASs** - Lowpass mode.

**Examples**
```
CALC:TRAN:FREQ BPAS
```

**Query Syntax**
```
CALCulate[:SELected]:TRANsform:FREQuency[:TYPE]?
```

**Return Type**
Character

**Default**
BPASs

---

**CALCulate[:SELected]:TRANsform:TIME:CENTer <num>**

*(Read-Write)* Set and query the center time for time domain measurements.

**Relevant Modes**
NA

**Parameters**
- `<num>`
  - Center time in seconds. Choose any number between: ± (number of points-1) / frequency span

**Examples**
```
CALC:TRAN:TIME:CENT 1e-9
```

**Query Syntax**
```
CALCulate[:SELected]:TRANsform:TIME:CENTer?
```

**Return Type**
Numeric

**Default**
0

---

Last Modified:
24-Jan-2012 New command
CALCulate[:SELected]:TRANsform:TIME:IMPulse:WIDth <num>

(Read-Write) Set and query the impulse width for the transform window.

Relevant Modes  NA

Parameters

<num> Impulse width in seconds; Choose any number between: .6 / frequency span and 1.39 / frequency span

Examples  CALC:TRAN:TIME:IMP:WIDT 10

Query Syntax  CALCulate[:SELected]:TRANsform:TIME:IMPulse:WIDth?

Return Type  Numeric

Default  .98 / Default Span

Last Modified:  24-Jan-2012 New command

CALCulate[:SELected]:TRANsform:TIME:KBESsel <num>

(Read-Write) Set and query the width for the Kaiser Bessel window.

Relevant Modes  NA

Parameters

<num> Window width for Kaiser Bessel in seconds; Choose any number between 0.0 and 13.0

Examples  CALC:TRAN:TIME:KBES 10

Query Syntax  CALCulate[:SELected]:TRANsform:TIME:KBESsel?

Return Type  Numeric

Default  6

Last Modified:  24-Jan-2012 New command

CALCulate[:SELected]:TRANsform:TIME:LPFRequency

(Write-only) Sets the start frequencies in LowPass Mode.

Relevant Modes  NA

Parameters  None
### Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:LPFR</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### Query Syntax

- **Calculation**: `CALCulate[:SELectioned]:TRANsform:TIME:SPAN?`
- **Return Type**: Numeric
- **Default**: 20 ns

### Last Modified:

- **24-Jan-2012**: New command

### Calculation[:SELectioned]:TRANsform:TIME:SPAN <num>

**Syntax**: `CALCulate[:SELectioned]:TRANsform:TIME:SPAN <num>`

**(Read-Write)** Set and query the span (stop - start) time for time domain measurements. Use `CALCulate[:SELectioned]:TRANsform:TIME:CENTer` to set the center time.

**Relevant Modes**: NA

**Parameters**

- `<num>`: Span time in seconds. Choose any number between: 0 and 2 * [(number of points-1) / frequency span].

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:SPAN 1e-9</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:SPAN 1e-9</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### Last Modified:

- **24-Jan-2012**: New command

### Calculation[:SELectioned]:TRANsform:TIME:STARt <num>

**Syntax**: `CALCulate[:SELectioned]:TRANsform:TIME:STARt <num>`

**(Read-Write)** Set and query the start time for time domain measurements. Use `CALCulate[:SELectioned]:TRANsform:TIME:STOP` to set the stop time.

**Relevant Modes**: NA

**Parameters**

- `<num>`: Start time in seconds. Choose any number between: ± (number of points-1) / frequency span.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:STARt 1e-9</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:STARt 1e-9</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC:TRAN:TIME:STARt 1e-9</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### Last Modified:

- **24-Jan-2012**: New command
CALCulate[:SELected]:TRANSform:TIME:STATe <bool>

(Read-Write) Set and query the ON |OFF Time Domain transform state for the active trace. Select a trace using CALCulate:PARameter:SELect.

Relevant Modes NA

Parameters

<num> Transform state. Choose from:

ON (or 1) - turns time domain ON.
OFF (or 0) - turns time domain OFF.

Examples CALC:TRAN:TIME:STAT 1

Query Syntax CALCulate[:SELected]:TRANSform:TIME:STATe?

Return Type Boolean

Default OFF (or 0).

Last Modified:
24-Jan-2012 New command

CALCulate[:SELected]:TRANSform:TIME:STEP:RTIMe <num>

(Read-Write) Set and query the step rise time for the transform window.

Relevant Modes NA

Parameters

<num> Rise time in seconds. Choose any number between:

.45 / frequency span and 1.48 / frequency span

Examples CALC:TRAN:TIME:STEP:RTIM 1e-8

Query Syntax CALCulate[:SELected]:TRANSform:TIME:STEP:RTIMe?

Return Type Numeric

Default .99 / Default Span

Last Modified:
24-Jan-2012 New command
**CALCulate[:SELeected]:TRANsform:TIME:STIMulus <char>**

*(Read-Write)* Set and query the type of simulated stimulus that will be incident on the DUT.

**Relevant Modes** NA

**Parameters**

<char> Choose from:

- **STEP** - simulates a step DUT stimulus (will automatically set `CALC:TRAN:TIME:TYPE` to LPASs)
- **IMPulse** - simulates a pulse DUT stimulus.

**Examples** CALC:TRAN:TIME:STIM STEP

**Query Syntax** CALCulate[:SELeected]:TRANsform:TIME:STIMulus?

**Return Type** Character

**Default** IMPulse

---

**CALCulate[:SELeected]:TRANsform:TIME:STOP <num>**

*(Read-Write)* Set and query the stop time for time domain measurements. Use `CALCulate[:SELeected]:TRANsform:TIME:STARt` to set the start time.

**Relevant Modes** NA

**Parameters**

<num> Stop time in seconds. Choose any number between: ± (number of points-1) / frequency span

**Examples** CALC:TRAN:TIME:STOP 1e-9

**Query Syntax** CALCulate[:SELeected]:TRANsform:TIME:STOP?

**Return Type** Numeric

**Default** 10 ns

---

**CALCulate[:SELeected]:TRANsform:TIME:TYPE <char>**
Set and query the transform type.

**Relevant Modes**  NA

**Parameters**

<num> Type of measurement. Choose from:

- **LPASs** - Lowpass; Must also send `CALC:TRAN:TIME:LPFRequency` before calibrating.
- **BPASs** - Bandpass; can only be used when `CALC:TRAN:TIME:STIM` is set to IMPulse.

**Examples**  CALC:TRAN:TIME:STEP:TYPE LPAS

**Query Syntax**  CALCulate[:SELected]:TRANsform:TIME:TYPE?

**Return Type**  Character

**Default**  BPASs

---

**CALibration:ZERO:TYPE:EXT**

(Write-Only) Performs external power meter zeroing.

**Relevant Modes**  Power meter

**Parameters**  None

**Examples**  CAL:ZERO:TYPE:EXT

**Query Syntax**  Not applicable

**Default**  Not applicable

---

**:CHSCanner:DATA?**

(Read) Query only, to return a set of comma-separated numeric values: id, center freq, integration BW, channel power.

**Relevant Modes**  Channel Scanner

**Parameters**  na

**Examples**  :CHSC :DATA?

**Query Syntax**  :CHSCanner:DATA?

**Return Type**  Comma-separated numeric

**Default**  Not Applicable
:CHSCanner:DISPlay:SORT

(Read-Write) Set and query the current sort by frequency and power value.

Relevant Modes  Channel Scanner

Parameters

Choose from:
Freq - Sort by frequency
Pow - Sort by power

Examples  :CHSC:DISP:SORT freq

Query Syntax  :CHSCanner:DISPlay:SORT?

Return Type

Default  Not Applicable

CHSCanner:DISPLAY:SORT:ORDer

(Read-Write) Set and query the current sort order up or down.

Relevant Modes  Channel Scanner

Parameters

Choose from:
Up - Sort frequencies or power values by ascending order
Down - Sort frequencies or power values by descending order

Examples  :CHSC:DISP:SORT:ORD up

Query Syntax  :CHSCanner:DISPlay:SORT:ORDer?

Return Type

Default  Not Applicable
:CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:PDIVsion

(Read-Write) Set and query the Y axis scale per division

**Relevant Modes**  Channel Scanner

**Parameters**

<val>Enter Y axis scale per division

**Examples**  :CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:PDIVsion 5

**Query Syntax**  :CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:PDIVsion?

**Return Type**

**Default**  Not Applicable

---

:CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:RLEVel

(Read-Write) Set and query the Y axis reference level

**Relevant Modes**  Channel Scanner

**Parameters**

<val>Enter Y axis reference level

**Examples**  :CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:RLEVel -20
              :CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:RLEVel 3

**Query Syntax**  :CHSCanner:DISPlay:WINdow:TRACe:Y[:SCALE]:RLEVel?

**Return Type**

**Default**  Not Applicable

---

:CHSCanner:EDIT:LIST?

(Read) Query the center frequency, frequency, span, and integration bandwidth

**Relevant Modes**  Channel Scanner
### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>n/a</th>
</tr>
</thead>
</table>

### Examples

<table>
<thead>
<tr>
<th>Examples</th>
<th>n/a</th>
</tr>
</thead>
</table>

### Query Syntax

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>:CHSCanner:EDIT:LIST?</th>
</tr>
</thead>
</table>

### Return Type

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Comma-separated numeric</th>
</tr>
</thead>
</table>

### Default

<table>
<thead>
<tr>
<th>Default</th>
<th>Not Applicable</th>
</tr>
</thead>
</table>

### Last Modified:

**:CHSCanner:EDIT:LIST:ADD**

**Write** Add a comma separated list that contains the following: center frequency (CF), frequency span (span), and integrated bandwidth (iBW)

**Relevant Modes** Channel Scanner

**Parameters**

Choose from:

- **cf** - sets the center frequency.
- **span** - sets the frequency span.
- **iBW** - sets the integrated bandwidth value.

**Examples**

<table>
<thead>
<tr>
<th>Examples</th>
<th>:CHSCanner:EDIT:LIST:ADD 1e9,10e6,3e6 'set center frequency to 1 GHz, span to 10 MHz, and the integrated bandwidth to 3 MHz</th>
</tr>
</thead>
</table>

**Query Syntax**

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>n/a</th>
</tr>
</thead>
</table>

**Return Type** Numeric

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Numeric</th>
</tr>
</thead>
</table>

### Default

<table>
<thead>
<tr>
<th>Default</th>
<th>Not Applicable</th>
</tr>
</thead>
</table>

### Last Modified:

**:CHSCanner:EDIT:LIST:CLEar**

**Write** Clears all items from the custom list.

**Relevant Modes** Channel Scanner

**Parameters** n/a
### :CHSCanner:EDIT:LIST:CLEar

<table>
<thead>
<tr>
<th>Examples</th>
<th>:CHSCanner:EDIT:LIST:CLEar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>n/a</td>
</tr>
<tr>
<td>Return Type</td>
<td>n/a</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Last Modified:

---

### :CHSCanner:EDIT:RANGe:COUNt

(Read-Write) Sets the total channel items in Range mode.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td><code>&lt;val&gt;</code> integer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th>:CHSCanner:EDIT:RANGe:COUNt 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>:CHSCanner:EDIT:RANGe:COUNt?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Last Modified:

---

### :CHSCanner:EDIT:RANGE:IBW

(Read-Write) Sets the value of the integration bandwidth.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th>:CHSCanner:EDIT:RANGE:IBW 2MHZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>:CHSCanner:EDIT:RANGE:IBW?</td>
</tr>
<tr>
<td>Return Type</td>
<td>numeric</td>
</tr>
<tr>
<td>Default</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Last Modified:
### :CHSCanner:LOG:ACTion:RECord

**(Write)** Start the recording process.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>Examples</td>
<td>:CHSCanner:LOG:ACTion:RECord</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>n/a</td>
</tr>
<tr>
<td>Return Type</td>
<td>n/a</td>
</tr>
<tr>
<td>Default</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Last Modified:

### :CHSCanner:LOG:ACTion:STOP

**(Write)** Stops the recording process.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>Examples</td>
<td>:CHSCanner:LOG:ACTion:STOP</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>n/a</td>
</tr>
<tr>
<td>Return Type</td>
<td>n/a</td>
</tr>
<tr>
<td>Default</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Last Modified:

### :CHSCanner:LOG:CONFigure:INTerval:DISTance

**(Read-Write)** Sets the value of the distance interval.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>Examples</td>
<td>:CHSCanner:LOG:CONFigure:INTerval:DISTance 5 'sets the Distance value to 5 m</td>
</tr>
<tr>
<td></td>
<td>:CHSCanner:LOG:CONFigure:INTerval:DISTance 4e3 'sets the</td>
</tr>
</tbody>
</table>
Distance value to 4 km

Query Syntax :CHSCanner:LOG:CONFigure:INTerval:DISTance?
Return Type numeric
Default n/a

Last Modified:

:CHSCanner:LOG:CONFigure:INTerval:STATe

(Read-Write) Enable or disables the measurement interval.
Relevant Modes Channel Scanner
Parameters
<bool> Choose from:
Off or 0 - Disable the measurement interval
On or 1 - Enable the measurement interval (1 km to 10 km)
Examples :CHSCanner:LOG:CONFigure:INTerval:STATe Off 'Disables the measurement interval
:CHSCanner:LOG:CONFigure:INTerval:STATe 1 'Enables the measurement interval
Query Syntax :CHSCanner:LOG:CONFigure:INTerval:STATe?
Return Type numeric
Default ON

Last Modified:

:CHSCanner:LOG:CONFigure:INTerval:TIME

(Read-Write) Sets the time interval (1 to 3,000 seconds).
Relevant Modes Channel Scanner
Parameters
<val>
Examples :CHSCannerLOG:CONFigure:INTerval:TIME 5
Query Syntax :CHSCannerLO:CONFigure:INTerval:TIME?
Return Type numeric
Command Reference

---

**Default**

---

**Last Modified:**

---

`:CHSCanner:LOG:CONFigure:INTerval:TYPe`

*(Read-Write)* Sets the time interval (1 to 3,000 seconds).

**Relevant Modes** Channel Scanner

**Parameters**

Choose from:

- **time** - sets the measurement interval type to time.
- **dist** - sets the measurement interval type to distance.

**Examples**

- `:CHSCanner:LOG:CONFigure:INTerval:TYPe time`
- `:CHSCanner:LOG:CONFigure:INTerval:TYPe dist`

**Query Syntax**

`:CHSCanner:LOG:CONFigure:INTerval:TYPe?`

**Return Type** character

---

**Last Modified:**

---

`:CHSCanner:LOG:CONFigure:SAVe:AUTo[:STATe]`

*(Read-Write)* Enables or disables the log file auto-save.

**Relevant Modes** Channel Scanner

**Parameters**

Choose from:

- **ON** - enables the log file auto-save.
- **OFF** - disables the log file auto-save.

**Examples**

- `:CHSCanner:LOG:CONFigure:SAVe:AUTo ON`

**Query Syntax**

`:CHSCanner:LOG:CONFigure:SAVe:AUTo?`

**Return Type** character

---

**Default**

---

**Last Modified:**
### :CHSCanner:LOG[:STATe]

**Read-Write** Sets the time interval (1 to 3,000 seconds).

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Choose from:</td>
<td></td>
</tr>
<tr>
<td><strong>STOP</strong> - stops the data logging.</td>
<td></td>
</tr>
<tr>
<td><strong>RUN</strong> - runs the data logging.</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>:CHSCanner:LOG STOP</td>
</tr>
<tr>
<td></td>
<td>:CHSCanner:LOG RUN</td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td>:CHSCanner:LOG[:STATe]?</td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>character</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>RUN</td>
</tr>
</tbody>
</table>

---

### :CHSChannel:SEARch:COUNt

**Read-Write** Select 1 of $n$ to be the active channel.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>Channel Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>$&lt;\text{val}&gt;$</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>:CHSChannel:SEARch:COUNt 3</td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td>:CHSChannel:SEARch:COUNt?</td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>numeric</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

### :CHSCanner:SEARch:TYPe

**Read-Write** Select top or bottom of $n$ total channels.
Relevant Modes  Channel Scanner
Parameters
Choose from:
top  - sets the channels to top down.
bottom  - sets the channels to bottom up.
Examples
:CHSCanner:SEARch:TYPE top,3
:CHSCanner:SEARch:TYPE bottom,10
Query Syntax
:CHSCanner:SEARch:TYPE?
Return Type  numeric
Default

Last Modified:

:CHSCanner[:SENSe]:CHANnel:SELect
(Read-Write) Select 1 to n to be the active channel.
Relevant Modes  Channel Scanner
Parameters
<val>
Examples
:CHSCanner[:SENSe]:CHANnel:SELect 3
Query Syntax
:CHSCanner[:SENSe]:CHANnel[:SELect]?
Return Type  numeric
Default

Last Modified:

:CHSCanner[:SENSe]:POWer[:RF]:ATTenuation
(Read-Write) Set the RF attenuator manually to a value of attenuation (5 to 30 dB).
Relevant Modes  Channel Scanner
Parameters
<val>0 to 30 dB
Examples
:CHSCanner[:SENSe]:POWer[:RF]:ATTenuation 15
Query Syntax
:CHSCanner[:SENSe]:POWer[:RF]:ATTenuation?
Return Type  numeric  
Default  10

Last Modified:

07sept2017  Updated parameter description.

:CHANnel[:SENSe]:POWer[:RF]:ATTenuation:AUTO

(Read-Write) Set the RF attenuator manually to automatic.

Relevant Modes  Channel Scanner

Parameters

Examples  :CHANnel[:SENSe]:POWer[:RF]:ATTenuation:AUTO

Query Syntax  :CHANnel[:SENSe]:POWer[:RF]:ATTenuation?

Return Type  numeric  
Default  10

Last Modified:

:CHANnel[:SENSe]:POWer[:RF]:EXTGain

(Read-Write) Set the external gain (G).

Relevant Modes  Channel Scanner

Parameters

Examples  :CHAN:POW:EXTG 3 'Sets the External Gain value to 3 dB.'

Query Syntax  :CHANnel[:SENSe]:POWer[:RF]:EXTGain?

Return Type  numeric

Default

Last Modified:

:CHANnel[:SENSe]:POWer[:RF]:GAIN[:STATe]
(Read-Write) Set the external gain (G).

**Relevant Modes** Channel Scanner

**Parameters**

<bool>Choose from:

**OFF** or **0** - disables the external gain

**ON** or **1** - enables the external gain

**Examples**

:CHSCanner[:SENSe]:POWer[:RF]:GAIN[:STATe] OFF

:CHSCanner[:SENSe]:POWer[:RF]:GAIN[:STATe] 1

**Query Syntax**

:CHSCanner[:SENSe]:POWer[:RF]:GAIN[:STATe]?

**Return Type** boolean

**Default**

Last Modified:

---

:CHSCanner:SWEep:AVERage:COUNt

(Read-Write) Set the averaging count.

**Relevant Modes** Channel Scanner

**Parameters**

<val>

**Examples**

:CHSCanner:SWEep:AVERage:COUNt 10

**Query Syntax**

:CHSCanner:SWEep:AVERage:COUNt?

**Return Type** numeric

**Default**

Last Modified:

---

:CHSCanner:SWEep:AVERage[:STATe]

(Read-Write) Enables or disables the averaging count.

**Relevant Modes** Channel Scanner

**Parameters**

<bool>Choose from:

**OFF** or **0** - disables the averaging count
**ON or 1 - enables the averaging count**

**Examples**

```
:CHSCanner:SWEep:AVERage[:STATe] ON
:CHSCanner:SWEep:AVERage[:STATe] 0
```

**Query Syntax**

```
:CHSCanner:SWEep:AVERage[:STATe]?
```

**Return Type** boolean

**Default**

---

**Last Modified:**

---

**:CHSCanner:SWEep:DISPlay:TYPE**

*(Read-Write)* Sets the display type.

**Relevant Modes** Channel Scanner

**Parameters**

```
<val>
```

Choose from:

- **BARC** - bar chart vertical
- **BHOR** - bar chart horizontal
- **CPOW** - channel
- **SCH** - strip chart
- **OVER** - strip chart overlay
- **SAL** - scan and listen

**Examples**

```
:CHSCanner:SWEep:DISPlay:TYPE BARC
:CHSCanner:SWEep:DISPlay:TYPE OVER
```

**Query Syntax**

```
:CHSCanner:SWEep:DISPlay:TYPE?
```

**Return Type** character

**Default**

---

**Last Modified:**

---

**:CHSCanner:SWEep:MODE**

*(Read-Write)* Sets the sweep mode.

**Relevant Modes** Channel Scanner

**Parameters**
<val>Choose from:

- **RANG** - Range
- **CLIST** - Custom List

**Examples**

:CHSCanner:SWEep:MODE **RANG**
:CHSCanner:SWEep:MODE **CLIST**

**Query Syntax**

:CHSC:USER:FOLD?

**Return Type** character

**Default**

---

**:CHSCanner:USER:FOLDer**

*(Read-Write)* Enables more flexibility where the Channel Scanner files are saved.

**Relevant Modes** Channel Scanner

**Parameters**

<val>Choose from:

- **DEF** - Default where:
  - List files saved-recalled to-from "CurrentDevice\ChannelScanner\ChannelLists"
  - Rec-Play files saved-recalled to-from "CurrentDevice\ChannelScanner\RecordPlayback"
- **SYST** - System where:
  - List files saved-recalled to-from "CurrentDevice\UserSelectedFolder"
  - Rec-Play files saved-recalled to-from "CurrentDevice\UserSelectedFolder"

**Examples**

:CHSC:USER:FOLD **SYST**

**Query Syntax**

:CHSC:USER:FOLD?

**Return Type** character

**Default**

---

**[:SENSe]:SAListen:DTYPE**

*(Read)* Sets the D type.

---

Last Modified:
Relevant Modes  Channel Scanner
Parameters

<val> Choose from:

AM- Amplitude modulation. In Tune & Listen, the demodulation bandwidth is narrower.

F
M- Frequency Modulation narrowband. In Tune & Listen, the demodulation bandwidth is narrower.

FMw- Frequency Modulation wideband. In Tune & Listen, the demodulation bandwidth is wider.

Examples

[SENSe]:SALiSTen:DTypE AM

[SENSe]:SALiSTen:DTypE FMw

Query Syntax

Return Type character

Default

Last Modified:

[:SENSe]:SALiSTen:LTImE

(Read) Sets the length of time Tune & Listen. i.e., How long to listen before jumping to the next frequency in the sequence.

Relevant Modes  Channel Scanner
Parameters

<val>

Examples

:SAL:LTIM 5

Query Syntax

Return Type character

Default

Last Modified:

[:SENSe]:SALiSTen:PAUSe

(Read) Pauses the data recording.

Relevant Modes  Channel Scanner
### Parameters

<bool>

### Examples

[:SENSe]:SAListen:PAUSe

### Query Syntax

Return Type: boolean

Default

---

**SENSe:SAListen:RESume**

*(Read)* Resumes the data recording.

**Relevant Modes** Channel Scanner

**Parameters**

<bool>

**Examples** [:SENSe]:SAListen:RESume

**Query Syntax**

Return Type: boolean

Default

---

**:CONFigure?**

*(Query only)* Queries the IQA configuration (Spectrum or Waveform).

**Relevant Modes** IQA

**Parameters**

n/a

**Examples** [:CONF?]

**Query Syntax** :CONF?

Return Type: character

Default: SPECtrum

---

Last Modified:
### :CONFigure:SPECtrum

*(Write Only)* Sets the measurement to Spectrum.

- **Relevant Modes**: IQA
- **Parameters**
  - n/a
- **Examples**: `CONF:SPEC`
- **Query Syntax**: n/a
- **Return Type**: Character
- **Default**: n/a

**Last Modified:**
- 22Oct2017    New IQA mode (10.1x)

### :CONFigure:WAVeform

*(Write Only)* Sets the measurement to waveform with RF Envelope displayed.

- **Relevant Modes**: IQA
- **Parameters**
  - n/a
- **Examples**: `CONF:WAV`
- **Query Syntax**: n/a
- **Return Type**: Character
- **Default**: n/a

**Last Modified:**
- 22Oct2017    New IQA mode (10.1x)

### :DISPlay:ADEMod:METRics:FM:RESults:DATA?
(Query only) Queries the FM demodulation data values. Returns 8 doubles values in the following order: carrierPower, carrierFreq, modulationRate, sinad, thd, fmFreqDeviation, fmFreqDeviationPlus, fmFreqDeviationMinus

**Relevant Modes** SA

**Parameters**

<number>

**Examples**

:DISP:ADEM:METR:FM:RES:DATA?

**Query Syntax**

:DISP:ADEM:METR:FM:RES:DATA?

**Return Type** Numbers

**Default**

**Notes** If query results are not ready or if the error "test tone not found" is displayed, the returned results are all zeroes ("0.00000000E+00").

---

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)
**DISPlay:ANNotation:FREQuency <value>**

*(Read-Write)* Set and query the method used to annotate frequency on the screen.

**Relevant Modes**  
FOPS

**Parameters**

<value>  
Choose from the following:

- **CSPan** - Center and Span
- **SSTop** - Start and Stop

**Examples**  
**DISP:ANN:FREQ CSP**

**Query Syntax**  
DISPlay:ANNotation:FREQuency?

**Return Type**  
Character

**Default**  
CSPan

---

Last modified:

19-Mar-2014  
New command (A.07.50)

---

**DISPlay:BRIghtness <num>**

*(Read-Write)* Set and query the brightness of the FieldFox display.

**Relevant Modes**  
ALL

**Parameters**

<num>  
Display brightness. Choose a number from 0 to 100. A zero setting is dim, but still viewable.

**Examples**  
**DISP:BRIG 50**

**Query Syntax**  
DISPlay:BRIghtness?

**Return Type**  
Numeric

**Default**  
0

---

**DISPlay:DATE:FMT <char>**

*(Read-Write)* Set and query the format of the system date. Set the date using **SYST:DATE**.


**Relevant Modes**  ALL

**Parameters**

<char>  System date format. Choose from:
       Argument -- Example
       HIDE  - Date is not shown on screen.
       SHORT  - 10/17/2008
       LONG  - Fri, 17 Oct 2008
       FULL  - Friday, October 17, 2008
       SORTABLE  - 2008-10-17
       MONTHDAY  - October 17

**Examples**  DISP:DATE:FMT Long

**Query Syntax**  DISP:DATE:FMT?

**Return Type**  Character

**Default**  Long

**DISPlay:ENABle <bool>**

(Read-Write)  Turns the FieldFox display ON and OFF. The OFF (0) setting causes faster SCPI operation.

**Relevant Modes**  ALL

**Parameters**

<bool>  Choose from:
       0 or OFF  - Display OFF. Send Preset or *RST to re-enable the display.
       1 or ON  - Display ON

**Examples**  DISP:ENAB 1

**Query Syntax**  DISP:ENABl?

**Return Type**  Boolean

**Default**  ON (1)

---

Last Modified:

20-Oct-2010  New command (5.30)

**DISPlay:GRID <bool>**

(Read-Write)  Set and query the visibility of the 10 x 10 grid on trace graph measurements.

**Relevant Modes**  USB PM: Pulse Measurements, FOPS

**Parameters**

<bool>  Choose from:
OFF or 0 - Grid OFF
ON or 1 - Grid ON

Examples

DISP:GRID 1

Query Syntax

DISPlay:GRID?

Default

OFF

Last modified:

19-Mar-2014 Added FOPS
31-Oct-2013 New command

DISPlay:KEYWord:DEFault

(Read-Only) Reset the FieldFox keywords to their default settings (listed below).

Relevant Modes

All

Parameters

None

Examples

DISPlay:KEYWord:DEFault

Query Syntax

Not Applicable

Default

"FILE","SITE","TOWER","TRACE","DATE","ANTENNA1","ANTENNA2","COMPANY1","COMPANY2","COM"

DISPlay:KEYWord[:DATA] <string1,2,3,4,5,6,7,8>

(Read-Write) Set and query the FieldFox keywords that can be used to create filenames.

Relevant Modes

All

Parameters

<string1,2,3,4,5,6,7,8> All 8 keywords, separated with commas, enclosed in individual quotes.

Examples

Disp:KEYW "Tower A","Tower B","Tower C","Tower D","Tower E","Tower F","Tower G","Tower H","Tower I","Tower J"

Query Syntax

DISPlay:KEYWord[:DATA]?

Return Type

String data, separated with commas, enclosed in individual quotes.

Default

See DISPlay:KEYWord:DEFault

:DISPlay:LTEFdd:BCHart[:Y[:SCALe]:AUTO

(Write Only) Autoscale the scale per division and reference values based on the current bar chart
measurement results.

**Relevant Modes**  
LTE FDD

**Parameters**

**Examples**  
DISP:LTEF:BCH:Y:AUTO

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

**DISPlay:LTEFdd:BCHart:Y[:SCALe]:PDIVision <num>**

(Read-Write) Set and query the scale-per-division value of the Bar Graph.

**Relevant Modes**  
LTE FDD

**Parameters**

<numeric>Minimum: 0.01 dB  
Maximum: 100 dB

**Examples**  
DISP:LTEF:BCH:Y:PDIV 10

**Query Syntax**  
DISP:LTEF:BCH:Y:PDIV?

**Default**  
10 db

---

**DISPlay:LTEFdd:BCHart:Y[:SCALe]:RLEVel**

(Read-Write) Set and query the trace reference level of the Bar Chart.

**Relevant Modes**  
LTE FDD

**Parameters**

<numeric>Minimum: -210 dBm  
Maximum: 100 dBm

**Examples**  
DISP:LTEF:BCH:Y:RLEV 20

**Query Syntax**  
DISP:LTEF:BCH:Y:RLEV?
**DISPlay:LTEFdd:SCHart:Y[:SCALE]:AUTO**

*(Write Only)* Autoscale the scale per division and reference values based on the current strip chart measurement results.

**Relevant Modes**  
LTE FDD

**Parameters**

**Examples**  
`DISP:LTEF:SCH:Y:AUTO`

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

**DISPlay:LTEFdd:SCHart:Y[:SCALE]:PDIVision**

*(Read-Write)* Set and query the scale-per-division value of the Strip Graph.

**Relevant Modes**  
LTE FDD

**Parameters**

- `<numeric>`
  - Minimum: 0.01 dB
  - Maximum: 100 dB

**Examples**  
`DISP:LTEF:SCH:Y:PDIV 10`

**Query Syntax**  
`DISP:LTEF:SCH:Y:PDIV?`

**Default**  
10 db

---

Last Modified:

06dec2018  
new command
**DISPlay:LTEFdd:SCHart:Y[:SCALe]:RLEVel**

(Read-Write) Set and query the trace reference level of the Strip Chart.

**Relevant Modes**  
LTE FDD

**Parameters**
- `<numeric>`
  - Minimum: -210 dBm
  - Maximum: 100 dBm

**Examples**
- `DISP:LTEF:SCH:Y:RLEV 15`

**Query Syntax**
- `DISP:LTEF:BSH:Y:RLEV?`

**Default**
- -10 dbm

---

**DISPlay:LTEFdd:SPECTrum:Y[:SCALe]:AUTO**

(Write Only) Autoscale the scale per division and reference values based on the current spectrum chart measurement results.

**Relevant Modes**  
LTE FDD

**Parameters**

**Examples**
- `DISP:LTEF:SPEC:Y:AUTO`

**Query Syntax**
- Not Applicable

**Default**
- Not Applicable

---

**DISPlay:LTEFdd:SPECTrum:Y[:SCALe]:PDIVision**

(Read-Write) Set and query the scale-per-division value of the Spectrum trace.

**Relevant Modes**  
LTE FDD
Parameters
<numeric> Minimum: 0.01 dB
          Maximum: 100 dB

Examples  DISP:LTEF:SPEC:Y:PDIV 100

Query Syntax  DISP:LTEF:SPEC:Y:PDIV?

Default  10 db

Last Modified:
06dec2018  new command

DISPlay:LTEFdd:SPECtrum:Y[:SCALe]:RLEVel

(Read-Write) Set and query the trace reference level of the Spectrum.

Relevant Modes  LTE FDD

Parameters
<numeric> Minimum: -210 dBm
          Maximum: 100 dBm

Examples  DISP:LTEF:SPEC:Y:RLEV 12

Query Syntax  DISP:LTEF:SPEC:Y:RLEV?

Default  -10 dbm

Last Modified:
06dec2018  new command

DISPlay:V5G:SPECtrum:Y[:SCALe]:RPOSition

(Read-Write) Set and query the trace reference position of the Spectrum.

Relevant Modes  5GTF

Parameters
<integer> Maximum: 10

Examples  DISP:V5G:SPEC:Y:RPOS 5

Query Syntax  DISP:V5G:SPEC:Y:RPOS?
Default  0

---

Last Modified:

06dec2018  new command

---

**:DISPlay:LTEFdd:WINDow<n>:DATA <string>*

(Read-Write) Set and query the selected window trace data.

**Relevant Modes**  LTE FDD

**Parameters**

<n>  Window number. If unspecified, value is set to 1.

n = 1 | 2 | 3 | 4  (Up to four windows can be selected/active one time on the FieldFox.)

<string>  Choices:

- **TABLE (TAB):** Cell scan numeric results (for up to 6 cell sites (ID's) including Physical Channel Information (PCI) – (Cell ID), RSRP, RSRQ, RSSI, PSS, SSS, SINR, Freq Err.
- **Bar CHart (BCH):** Vertical power bar graph of selectable cell scan result metric for up to 6 cell sites with adjustable color "heat" amplitude scale
- **SPECtrum (SPEC):** Magnitude spectrum frequency domain (fixed span). You can only view a single component carrier at a time with Spectrum (e.g., CC0 or CC2, etcetera).
- **Strip CHart (SCH):** Magnitude of selectable cell scan result metric graphed over time.

**Examples**

DISP:LTEF:WIND2:DATA BCH
DISP:LTEF:WIND4:DATA SPEC

**Query Syntax**  DISP:LTEF:WIND3:DATA?

**Return Type**  string

**Default**  TABL

---

Last Modified:

19-Dec-2018  New command

---

**:DISPlay:LTEFdd:WINDow<n>:CCARier <string>*
(Read-Write) Set and query the displayed component carrier (CC).

**Relevant Modes**  
**LTE FDD**

### Parameters

- `<n>` Window number. If unspecified, value is set to 1.  
  \[ n = 1 \mid 2 \mid 3 \mid 4 \]  
  (Up to four windows can be selected/active at one time on the FieldFox.)

- `<string>` Choices:  
  - **ALL**: All component carriers are displayed that meet the current setup’s measurement criteria  
  - **CC0:CC4**: Component carrier 0: component carrier 4

### Examples

- `DISP:LTEF:WIND2:CCAR ALL`
- `DISP:LTEF:WIND4:CCAR CC4`

### Query Syntax

`DISP:LTEF:WIND3:CCAR?`

**Return Type**  
string

**Default**  
ALL

---

**Last Modified:**  
19-Dec-2018  
New command

---

```plaintext
:DISPlay:LTEFdd:WINDow<n>:DATA <string>
```

(Read-Write) Set and query the selected window trace data.

**Relevant Modes**  
**LTE FDD**

### Parameters

- `<n>` Window number. If unspecified, value is set to 1.  
  \[ n = 1 \mid 2 \mid 3 \mid 4 \]  
  (Up to four windows can be selected/active one time on the FieldFox.)

- `<string>` Choices:  
  - **TABLe (TAB)**: Cell scan numeric results (for up to 6 cell sites (ID’s) including Physical Channel Information (PCI) – (Cell ID), RSRP, RSRQ, RSSI, PSS, SSS, SINR, Freq Err.
  - **Bar CHart (BCH)**: Vertical power bar graph of selectable cell scan result metric for up to 6 cell sites with adjustable color “heat” amplitude scale
  - **SPECtrum (SPEC)**: Magnitude spectrum frequency domain (fixed span). You can only view a single component carrier at a time with Spectrum (e.g., CC0 or CC2, etcetera).
  - **Strip CHart (SCH)**: Magnitude of selectable cell scan result metric graphed over time.

Couplings: Strip Chart should be active in only one window. If the strip chart
window is disabled, it can be re-enabled in another active window. When the strip chart window is re-enabled, trace data is automatically reset in the Table.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DISP:LTEF:WIND2:DATA BCH</code></td>
<td></td>
</tr>
<tr>
<td><code>DISP:LTEF:WIND4:DATA SPEC</code></td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax**

`DISP:LTEF:WIND3:DATA?`

**Return Type**

`string` (Default: `TABL`)

---

**Last Modified:**

19-Dec-2018 New command

---

**:DISPLAY:LTEFdd:WINDOW<n>:SORT:DATA <string>**

*(Read-Write)* Set and query the sorted data type.

See also `DISP:LTEF:WIND<n>:SORT:ORD`.

**Relevant Modes**

`LTE FDD`

**Parameters**

- `<n>` Window number. If unspecified, value is set to 1.
  - `n = 1 | 2 | 3 | 4` (Up to four windows can be selected/active one time on the FieldFox.)
- `<string>` Choices:
  - `RSRP`: Reference Symbol Received Power
  - `RSRQ`: Reference Signal Received Quality
  - `RSSI`: Reference Signal Strength Indicator
  - `PSS`: The power level in dB for the primary synchronization signal, relative to the average subcarrier power.
  - `SSS`: The power level in dB for the secondary synchronization signal, relative to the average subcarrier power.
  - `SINR`: Reference value that is used in a system simulation. Subcarriers and or resource elements are measured over the same bandwidth.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DISP:LTEF:WIND2:SORT:DATA SINR</code></td>
<td></td>
</tr>
<tr>
<td><code>DISP:LTEF:WIND4:SORT:DATA PSS</code></td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax**

`DISP:LTEF:WIND3:SORT:DATA?`

**Return Type**

`string` (Default: `RSRP`)

---

**Last Modified:**

19-Dec-2018 New command
**:DISPlay:LTEFdd:WINdow<n>:SORT:ORDer <string>**

*(Read-Write)* Set and query the data's sort sequence.
See also **DISP:LTEF:WINdow<n>:SORT:DATA**.

**Relevant Modes**  
LTE FDD

**Parameters**

<n>  
Window number. If unspecified, value is set to 1.  
\( n = 1 \mid 2 \mid 3 \mid 4 \) (Up to four windows can be selected/active one time on the FieldFox.)

<string>  
Choices:  
AUTO: Displayed scan results is not sorted and had no related sequence.  
UP: Displayed scan results is sorted up.  
DOWN: Displayed scan results is sorted down.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DISP:LTEF:WIN2:SORT:ORD AUTO</strong></td>
</tr>
<tr>
<td><strong>DISP:LTEF:WIN4:SORT:ORD UP</strong></td>
</tr>
</tbody>
</table>

**Query Syntax**  
**DISP:LTEF:WIN3:SORT:ORD?**

**Return Type** string

**Default** AUTO

---

Last Modified:

19-Dec-2018  New command

---

**:DISPlay:LTEFdd:TRACe:DATA <string>**

*(Read-Write)* Set and query the trace data type (Bar Chart and Strip Chart only).

**NOTE:** Only one trace data type result can be displayed. Choice decides which data type is shown on GUI.

**Relevant Modes**  
LTE FDD

**Parameters**

<string>  
Choices:  
RSRP: Reference Symbol Received Power  
RSRQ: Reference Signal Received Quality  
RSSI: Reference Signal Strength Indicator  
PSS: The power level in dB for the primary synchronization signal, relative to the average subcarrier power.  
SSS: The power level in dB for the secondary synchronization signal, relative to the average subcarrier power.
**SINR:** Reference value that is used in a system simulation. Subcarriers and or resource elements are measured over the same bandwidth.

**Examples**
- `DISP:LTEF:TRACe:DATA SINR`
- `DISP:LTEF:TRACe:DATA PSS`

**Query Syntax**
- `DISP:LTEF:TRACe:DATA?`

**Return Type**
- string

**Default**
- RSRP

---

**:DISPlay:LTEFdd:WINDow[:SELect] <int>**

*(Read-Write)* Set and query the selected window.

**Relevant Modes**
- LTE FDD

**Parameters**
- `<integer>`
  - Choices: 1 through 4. (Up to four windows can be selected/active at one time on the FieldFox.)

**Examples**
- `DISP:LTEF:WIND 2`

**Query Syntax**
- `DISP:LTEF:WIND?`

**Return Type**
- numeric

**Default**
- 1

---

**:DISPlay:LTEFdd:WINDow<n>:STATe <bool>**

*(Read-Write)* Set and query the selected window state.

**Relevant Modes**
- LTE FDD

**Parameters**
- `<n>`
  - Window number. If unspecified, value is set to 1.

  
  \(n = 1 \mid 2 \mid 3 \mid 4\)  
  - (Up to four windows can be selected/active at one time on the FieldFox.)

- `<boolean>`
  - **ON (1):** Enable the extended frequency range.
OFF (2): Disable the extended frequency range

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP:LTEF:WIND2:STAT ON</td>
<td>Enable extended frequency range for WIND2</td>
</tr>
<tr>
<td>DISP:LTEF:WIND4:STAT 0</td>
<td>Disable extended frequency range for WIND4</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP:LTEF:WIND3:STAT?</td>
<td>Query extended frequency range state for WIND3</td>
</tr>
</tbody>
</table>

**Query Syntax**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP:LTEF:WIND3:STAT?</td>
<td>Query extended frequency range state for WIND3</td>
</tr>
</tbody>
</table>

**Return Type**

- numeric

**Default**

- 1

---

**DISPlay:MARKer:LARGe:STATe <char>**

**(Read-Write)** Set and query the display state of large marker readouts.

**Relevant Modes**

- NA

**Parameters**

- `<char>` Large marker readout state. Choose from:
  - OFF - No large marker readout.
  - A - The A display state is recalled. If none have been defined, then the default A display state is recalled.
  - B - The B display state is recalled. If none have been defined, then the default B display state is recalled.

**Examples**

```plaintext
DISP:MARK:LARG:STAT A
```

**Query Syntax**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP:MARK:LARG:STAT?</td>
<td>Query large marker display state</td>
</tr>
</tbody>
</table>

**Return Type**

- Character

**Default**

- OFF

---

**DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:FORMat <char>**

**(Read-Write)** Set and query the format for the specified readout line on the large marker display state.

**Relevant Modes**

- NA

**Parameters**

- `<x>` Large Marker Display State to edit.
Choose from: A or B

Note: This is a SCPI node and not an argument. See example below.

<n> Readout line number. Choose from 1, 2, or 3.
<char> Large marker readout format. Choose from:
  DEFault - same format as the trace on which the marker resides.
  MAGPhase - magnitude and phase
  IMPedance - Complex impedance format: R + jX
  ZMAGntude - Impedance; magnitude only
  PHASE
  REAL
  IMAGinary
  FREQuency - displays ONLY the frequency of the marker

Examples

```
DISP:MARK:LARG:A:DEF:LINE1:FORM phase
```

Query Syntax

```
DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:FORMat?
```

Return Type

Character

Default

DEFault

Last Modified:

22-Sept-2014 New command

---

```
DISPlay:MARKer:LARGe:<x>:DEFine:LINE:MARKer<n>:STATe <char>
```

(Read-Write) Set and query how the specified marker is displayed.

Relevant Modes

NA

Parameters

- <x> Display state. Choose from:
  - A - Display state A
  - B - Display state B
- <n> Marker number. Choose from:
  - 1, 2, or 3
- <char> Marker display state. Choose from:
  - OFF - Specified marker is set OFF.
  - NORMal - Specified marker is a normal marker.
  - DELTa - Specified marker is a delta marker.

Examples

```
```

Query Syntax

```
DISPlay:MARKer:LARGe:<x>:DEFine:LINE:MARKer<n>:STATe?
```
Return Type: Character
Default: NORMal

Last Modified:
8-Sept-2014 New command

**DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:MNUMber <value>**

(Read-Write) Set and query the marker number to assign to the specified marker readout line.

**Relevant Modes** NA

**Parameters**

- `<x>` Large Marker Display State to edit.
  Choose from: A or B

  **Note:** This is a SCPI node and not an argument. See example below.

- `<n>` Readout line number. Choose from 1, 2, or 3.

- `<value>` Marker number. Choose a marker from 1 through 6.

**Examples**

```
```

**Query Syntax**

DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:MNUMber?

**Return Type** Numeric

**Default** Marker 1 is assigned to all three readouts, different traces.

Last Modified:
22-Sept-2014 New command

**DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:STATe <bool>**

(Read-Write) Set and query the ENABLE state for the specified readout line.

**Relevant Modes** NA

**Parameters**

- `<x>` Large Marker Display State to edit.
  Choose from: A or B

  **Note:** This is a SCPI node and not an argument. See example below.

- `<n>` Readout line number. Choose from 1, 2, or 3.
**<bool>**  Enable state. Choose from:
- **OFF** or **0** - Specified readout line is OFF.
- **ON** or **1** - Specified readout line is ON.

**Examples**  
```
DISP:MARK:LARG:A:DEF:LINE1:STAT 1
```

**Query Syntax**  
```
DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:STATe?
```

**Return Type**  
Boolean

**Default**  
Readout line 1 is ON when the display state is first recalled. Readout lines 2 and 3 are OFF when the display state is first recalled.

---

**DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:TNUMber <value>**

*(Read-Write)*  
Set and query the trace number to assign to the specified marker readout line.

**Relevant Modes**  
**NA**

**Parameters**

- **<x>**  
  Large Marker Display State to edit. Choose from: **A** or **B**
  
  **Note:** This is a SCPI node and not an argument. See example below.

- **<n>**  
  Readout line number. Choose from 1, 2, or 3.

- **<value>**  
  Trace number. Choose a trace from 1 through 3.

**Examples**  
```
```

**Query Syntax**  
```
DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:TNUMber?
```

**Return Type**  
Numeric

**Default**  
Traces 1, 2, and 3 are assigned to the corresponding readouts (1, 2, 3)

---

**DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:TRACking <bool>**

*(Read-Write)*  
Set and query the marker tracking state for the specified readout line.

Select the parameter to track on the standard Marker Search menu.

When set to **ON**, Peak tracking is set by default.
To set a different marker search function, use the standard `Calc:Mark:Function` commands.

**Relevant Modes** NA

**Parameters**

- `<x>` Large Marker Display State to edit. Choose from: A or B
  
  **Note:** This is a SCPI node and not an argument. See example below.

- `<n>` Readout line number. Choose from 1, 2, or 3.

- `<bool>` Marker tracking state. Choose from:
  
  - OFF or 0 - Tracking disabled.
  - ON or 1 - Tracking enabled.

**Examples**

- `DISP:MARK:LARG:A:DEF:LINE1:TRACking 1`

**Query Syntax**

`DISPlay:MARKer:LARGe:<x>:DEFine:LINE<n>:TRACking?`

**Return Type** Boolean

**Default** OFF

-------------------------------

**Last Modified:**

22-Sept-2014 New command

---

**DISPlay:MARKer:LARGe:<x>:DEFine:TRACe<n>:BWIDth:STATe <value>**

*(Read-Write)* Set and query the marker bandwidth search state for the specified trace.

**Relevant Modes** NA

**Parameters**

- `<x>` Large Marker Display State to edit. Choose from: A or B
  
  **Note:** This is a SCPI node and not an argument. See example below.

- `<n>` Trace number. Choose from 1, 2, or 3.

- `<value>` Marker bandwidth search state. Choose from:
  
  - OFF or 0 - BW search is OFF.
  - ON or 1 - BW search is ON.

**Examples**


**Query Syntax**

`DISPlay:MARKer:LARGe:<x>:DEFine:TRACe<n>:BWIDth:STATe?`

**Return Type** Boolean

**Default** OFF
DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:FORMat <value>

(Read-Write) Set and query the display format for the specified trace.

**Parameters**

- `<x>` Large Marker Display State to edit. Choose from: A or B
  
  **Note:** This is a SCPI node and not an argument. See example below.

- `<n>` Trace number. Choose from 1, 2, or 3.

- `<value>` Display format. Choose from:
  - MLOGarithmic – Log magnitude
  - MLINear – Linear magnitude
  - SWR – Standing Wave Ratio
  - PHASE - Phase in degrees. The trace wraps every 360 degrees, from +180 to –180
  - UPHase - Unwrapped phase in degrees.
  - SMITH – Smith chart; series resistance and reactance.
  - POLar - Magnitude and phase of the reflection coefficient.
  - GDELAY – Group delay

**Examples**


**Query Syntax**

DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:FORMat?

**Return Type**

Character

**Default**

MLOG

---

DISPlay:MARKer:LARGE:<x>:DEFine:TRACe<n>:MEASurement <value>

(Read-Write) Set and query the measurement to display for the specified trace.
Relevant Modes  NA

Parameters

<x> Large Marker Display State to edit.
Choose from: A or B

Note: This is a SCPI node and not an argument. See example below.

<n> Trace number. Choose from 1, 2, or 3.

<value> Measurement. Choose from:
- S11 - Forward reflection measurement
- S21 - Forward transmission measurement
- S12 - Reverse transmission
- S22 - Reverse reflection
- A - A receiver measurement
- B - B receiver measurement
- R1 - Port 1 reference receiver measurement
- R2 - Port 2 reference receiver measurement


Query Syntax  DISPLAY:MARKer:LARGe:<x>:DEFine:TRACe<n>:MEASurement?

Return Type  Character

Default  Trace1=S11
Trace2=S21
Trace1=S12

Last Modified:
22-Sept-2014  New command

DISPLAY:MARKer:LARGe:<x>:FONT <char>

(Read-Write) Set and query the font size for the large marker readouts.

Relevant Modes  NA

Parameters

<x> Large Marker Display State to edit.
Choose from: A or B

Note: This is a SCPI node and not an argument. See example below.

<char> Font size. Choose from:
BIG - Big size.
**SUP** - Super big size.

**Examples**

```
DISP:MARK:LARG:A:FONT SUP
```

**Query Syntax**

```
DISPLAY:MARKer:LARGe:<x>:FONT?
```

**Return Type**

Character

**Default**

BIG

---

**DISPlay:MARKer:LARGe:<x>:TRACe:COUNt <char>**

*(Read-Write)* Set and query the number of traces to include in the large marker display state.

**Relevant Modes**

NA

**Parameters**

- `<x>`: Display state. Choose from:
  - A: Display state A
  - B: Display state B

- `<char>`: Trace configuration. Choose from:
  - D1: One trace.
  - D2: Two traces overlaid on one graticule.
  - D3: Three traces overlaid on one graticule.

**Examples**

```
DISP:MARK:LARG:B:TRAC:COUN D3
```

**Query Syntax**

```
DISPLAY:MARKer:LARGe:<x>:TRACe:COUNt?
```

**Return Type**

Character

**Default**

D1

---

**DISPlay:SCReen:GEOMetry <char>**

*(Read-Write)* Set and query the visibility of the 10 x 10 grid on trace graph measurements.

**Relevant Modes**

USB PM: Pulse Measurements, FOPS

**Parameters**
Choose from:
**NORM** - Normal (with softkeys)
**FULL** - Full screen (without softkeys)

**Examples**
:DISP:SCR:GEOM NORM

**Query Syntax**
DISPLAY:SCR:GEOM?

**Default**
NORM

Last modified:
26-Apr-2016 New command

**DISPLAY:TABLE:MARKer <bool>**

*(Read-Write)* Set and query the display of the marker table.

**Relevant Modes**
CAT, NA, RTSA, SA, **Pulse Measurements**

**Parameters**

< bool >  Marker table display state. Choose from:
O or OFF - Table OFF
1 or ON - Table ON

**Examples**
DISPLAY:TABLE:MARK ON

**Query Syntax**
DISPLAY:TABLE:MARK?

**Return Type**
Boolean

**Default**
OFF

Last modified:
08-Aug-2016 Added RTSA
31-Oct-2013 Added Pulse

**DISPLAY:TABLE:RESULTS:DATA?**

*(Read-Only)* Read the 10 Auto Analysis results.

**Relevant Modes**
**Pulse Measurements**

**Parameters**
None

**Examples**
DISPLAY:TABLE:RES:DATA?

**Default**
N/A
DISPlay:TABLE:RESults <bool>

(Read-Write) Set and query the visibility of the of both the Marker table and the Auto Analysis table.

Relevant Modes  Pulse Measurements

Parameters

<bool> Choose from:
OFF or 0 - Tables OFF
ON or 1 - Tables ON

Examples  DISP:TABLE:RES 1

Query Syntax  DISPlay:TABLE:RESults?

Default  OFF

DISPlay:TIME:FMT <char>

(Read-Write) Set and query the format of the system time on the screen. Set the time using SYST:TIME

Relevant Modes  ALL

Parameters

<char> System time format. Choose from:
Argument - Example
HIDE - time is not shown on screen.
SHORT - 10:31 PM
LONG - 10:31:25 PM
SH24 - 22:31
LO24 - 22:31:25

Examples  DISP:TIME:FMT Long

Query Syntax  DISPlay:TIME:FMT?

Return Type  Character

Default  Long
**DISPlay:TITLe:DATA <string>**

*(Read-Write)* Set and query the title that appears in the upper-left corner of the FieldFox screen. The title can contain up to approximately 65 alpha-numeric characters. Display the title using `DISPlay:TITLe:STATe`.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td><code>&lt;num&gt;</code> Title</td>
<td>Choose any string.</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td><code>DISP:TITL:DATA &quot;My title&quot;</code></td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td><code>DISPlay:TITLe:DATA?</code></td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>String</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>&quot;User Title&quot;</td>
</tr>
</tbody>
</table>

**DISPlay:TITLe[:STATe] <bool>**

*(Read-Write)* Set and query display state of the title string. Change the title using `DISPlay:TITLe`.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td><code>&lt;bool&gt;</code></td>
<td>Choose from:</td>
</tr>
<tr>
<td>O or OFF - Title OFF</td>
<td></td>
</tr>
<tr>
<td>1 or ON - Title ON</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td><code>DISP:TITL 1</code></td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td><code>DISPlay:TITLe[:STATe]?</code></td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>Boolean</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>0 - Off</td>
</tr>
</tbody>
</table>

**DISPlay:V5G:BCHart:Y[:SCALe]:AUTO**

*(Write Only)* Autoscale the scale per division and reference values based on the current bar chart measurement results.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>5GTF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td><code>DISP:V5G:BCH:Y:AUTO</code></td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td>Not Applicable</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>
**DISPlay:V5G:BCHart:Y[:SCALe]:PDIVision**

(Read-Write) Set and query the scale-per-division value of the Bar Graph.

- **Relevant Modes**: 5GTF
- **Parameters**
  - `<numeric>` Minimum: 0.01 dB
  - Maximum: 100 dB
- **Examples**
  - DISP:V5G:BCH:Y:PDIV 10
- **Query Syntax**
  - DISP:V5G:BCH:Y:PDIV?
- **Default**: 10 db

---

**DISPlay:V5G:BCHart:Y[:SCALe]:RLEVel**

(Read-Write) Set and query the trace reference level of the Bar Chart.

- **Relevant Modes**: 5GTF
- **Parameters**
  - `<numeric>` Minimum: -210 dBm
  - Maximum: 100 dBm
- **Examples**
  - DISP:V5G:BCH:Y:RLEV 20
- **Query Syntax**
  - DISP:V5G:BCH:Y:RLEV?
- **Default**: -10 dbm

---

**DISPlay:V5G:SCHart:Y[:SCALe]:AUTO**
(Write Only) Autoscale the scale per division and reference values based on the current strip chart measurement results.

**Relevant Modes** 5GTF

**Parameters**

**Examples** `DISP:V5G:SCH:Y:AUTO`

**Query Syntax** Not Applicable

**Default** Not Applicable

Last Modified:
06dec2018  new command

### DISPlay:V5G:SCHart:Y[:SCALe]:PDIVision

(Read-Write) Set and query the scale-per-division value of the Strip Graph.

**Relevant Modes** 5GTF

**Parameters**

<numeric>Minimum: 0.01 dB

Maximum: 100 dB

**Examples** `DISP:V5G:SCH:Y:PDIV 10`

**Query Syntax** `DISP:V5G:SCH:Y:PDIV?`

**Default** 10 db

Last Modified:
06dec2018  new command

### DISPlay:V5G:SCHart:Y[:SCALe]:RLEVel

(Read-Write) Set and query the trace reference level of the Strip Chart.

**Relevant Modes** 5GTF

**Parameters**

<numeric>Minimum: -210 dBm

Maximum: 100 dBm
### DISP:V5G:SCH:Y:RLEV

**Examples**

```
DISP:V5G:SCH:Y:RLEV 20
```

**Query Syntax**

```
DISP:V5G:SCH:Y:RLEV?
```

**Default**

-10 dbm

---

### DISP:V5G:SPEClrum:Y[:SCALe]:AUTO

**DISPlay:V5G:SPEClrum:Y[:SCALe]:AUTO**

*(Write Only)* Autoscale the scale per division and reference values based on the current spectrum chart measurement results.

**Relevant Modes**

- 5GTF

**Parameters**

**Examples**

```
DISP:V5G:SPEClrum:Y:AUTO
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

### DISP:V5G:SPEClrum:Y[:SCALe]:PDIVision

*(Read-Write)* Set and query the scale-per-division value of the Spectrum trace.

**Relevant Modes**

- 5GTF

**Parameters**

- `<numeric>` Minimum: 0.01 dB
  - Maximum: 100 dB

**Examples**

```
DISP:V5G:SPEClrum:Y:PDIV 5
```

**Query Syntax**

```
DISP:V5G:SPEClrum:Y:PDIV?
```

**Default**

10 db

---

Last Modified:

06dec2018 new command
### DISPLAY:V5G:SPECTrum:Y[:SCALe]:RLEVel

(Read-Write) Set and query the per division reference level of the Spectrum.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>5GTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;integer&gt; Maximum: 10</td>
</tr>
<tr>
<td>Examples</td>
<td>DISPV5G:SPEC:Y:RLEV 5</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>DISPV5G:SPEC:Y:RLEV?</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>

Last Modified:

06dec2018 new command

### DISPLAY:V5G:SPECTrum:Y[:SCALe]:RPOSition

(Read-Write) Set and query the trace reference position of the Spectrum.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>5GTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;integer&gt; Maximum: 10</td>
</tr>
<tr>
<td>Examples</td>
<td>DISPV5G:SPEC:Y:RPOS 5</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>DISPV5G:SPEC:Y:RPOS?</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>

Last Modified:

06dec2018 new command

### DISPLAY:V5G:WINDow<n>:CCARrier <string>

(Read-Write) Set and query the component carrier (CC).
Relevant Modes  
**5GTF**

Parameters

$<n>$  
Window number. If unspecified, value is set to 1.  
$n = 1 \mid 2 \mid 3 \mid 4$ (Up to four windows can be selected/active at one time on the FieldFox.)

$<\text{string}>$  
Choices:
- **ALL**: All component carriers are displayed that meet the current setup's measurement criteria
- **CC0:CC7**: Component carrier 0:component carrier 7

Examples
- `DISP:V5G:WIND2:CCAR ALL`
- `DISP:V5G:WIND4:CCAR CC7`

Query Syntax  
`DISP:V5G:WIND3:CCAR?`

Return Type  
string

Default  
ALL

---

Last Modified:  
19-Dec-2018  
New command

**DISPlay:**V5G:WINDow$<n>$:DATA $<\text{string}>$  
(Read-Write) Set and query the selected window data.

Relevant Modes  
**5GTF**

Parameters

$<n>$  
Window number. If unspecified, value is set to 1.  
$n = 1 \mid 2 \mid 3 \mid 4$ (Up to four windows can be selected/active one time on the FieldFox.)

$<\text{string}>$  
Choices:
- **TABLE (TAB)**: Cell scan numeric results (for up to 6 cell sites (ID's) including Physical Channel Information (PCI) – (Cell ID), PSS, SSS, Channel Power, Sync Correlation Threshold.
- **Bar CHart (BCH)**: Vertical power bar graph of selectable cell scan result metric for up to 6 cell sites with adjustable color "heat" amplitude scale
- **SPECtrum (SPEC)**: Magnitude spectrum frequency domain (fixed span). You can only view a single component carrier at a time with Spectrum (e.g., CC0 or CC2, etcetera).
- **Strip CHart (SCH)**: Magnitude of selectable cell scan result metric graphed over time.

Couplings: Strip Chart should be active in only one window. If the strip chart window is disabled, it can be re-enabled in another active window. When the
strip chart window is re-enabled, trace data is automatically reset to Table.

Examples
- DISP:V5G:WIND2:DATA BCH
- DISP:V5G:WIND4:DATA SPEC

Query Syntax
- DISP:V5G:WIND3:DATA?

Return Type
- string

Default
- TABL

Last Modified:
19-Dec-2018 New command
**DISPlay:V5G:WINDow<n>:SORT:ORDer <string>**

(Read-Write) Set and query the data's sort sequence.

See also **DISP:V5G:WIND<n>:SORT:DATA**.

**Relevant Modes** 5GTF

**Parameters**

<n> Window number. If unspecified, value is set to 1.

n = 1 | 2 | 3 | 4  (Up to four windows can be selected/active one time on the FieldFox.)

<string> Choices:

AUTO: Displayed scan results is not sorted and had no related sequence.

UP: Displayed scan results is sorted up.

DOWN: Displayed scan results is sorted down.

**Examples**

DISP:V5G:WIND2:SORT:ORD AUTO
DISP:V5G:WIND4:SORT:ORD UP

**Query Syntax** DISP:V5G:WIND3:SORT:ORD?

**Return Type** string

**Default** AUTO

---

**DISPlay:V5G:WINDow<n>:STATe**

(Read-Write) Set and query the selected window state.

**Relevant Modes** 5GTF

**Parameters**

<n> Window number. If unspecified, value is set to 1.

n = 1 | 2 | 3 | 4  (Up to four windows can be selected/active at one time on the FieldFox.)

<boolen> Choices:

ON (1): Enable the window.

OFF (2): Disable the window

**Examples**

DISP:V5G:WIND2:STAT ON
DISP:V5G:WIND4:STAT 0

**Query Syntax** DISP:V5G:WIND3:STAT?

**Return Type** numeric
DISPlay:V5G:TRACe:DATA <string>

(Read-Write) Set and query the trace data type (Bar Chart and Strip Chart only).

NOTE: Only one trace data type result can be displayed. Choice decides which data type is shown on GUI.

**Relevant Modes** 5GTF

**Parameters**

<string> Choices:

- **PSS**: The power level in dB for the primary synchronization signal, relative to the average subcarrier power.
- **SSS**: The power level in dB for the secondary synchronization signal, relative to the average subcarrier power.
- **POW**: Channel power (dBm).

**Examples**

- DISP:V5G:TRACe:DATA POW
- DISP:V5G:TRACe:DATA PSS

**Query Syntax** DISP:V5G:TRACe:DATA?

**Return Type** string

**Default** PSS

Last Modified:

19-Dec-2018 New command

DISPlay:WAVeform:VIEW[:SELect] <char>

(Read-Write) Set and query the selected displayed waveform type.

**Relevant Modes** IQA

**Parameters**

<character> Waveform type. Choose from:

- **RFEN** - RF envelope
**IQW** - I/Q waveform

**Examples**
```
DISP:WAV:VIEW IQW
```

**Query Syntax**
```
:DISP:WAV:VIEW?
```

**Return Type** Character

**Default** RFEN

---

Last Modified:

22-oct-2017 Added new IQA mode (10.1x).

---

**:DISPlay:VIEW:DENSity:STATe**

*(Read-Write)* Enables/Disables the RTSA density Show Density graphics. (Same as pressing the Show Density button.)

**Relevant Modes** RTSA

**Parameters**

- `<bool>` Show density graphics value:
  - 0 or 1

**Examples**
```
DISP:VIEW:DENS:STAT 1
DISP:VIEW:DENS:STAT 0
```

**Query Syntax**
```
DISP:VIEW:DENS:STAT?
```

**Return Type** bool

**Default** 1

---

Last modified:

08-Aug-2016 Added for RTSA Mode (A.09.50)

---

**:DISPlay:VIEW:DENSity:PERSistence**

*(Read-Write)* Set and query the RTSA density persistence value.

**Relevant Modes** RTSA

**Parameters**

- `<num>` Persistence value:
  - 0 to 5 seconds

**Examples**
```
DISP:VIEW:DENS:PERS 1
DISP:VIEW:DENS:PERS .002
```

**Query Syntax**
```
DISP:VIEW:DENS:PERS?
```
Return Type  alpha-numeric
Default  0.000E+00

Last modified:
08-Aug-2016  Added for RTSA Mode (A.09.50)


(Read-Write) Enables/Disables the RTSA density persistence infinite value.

Relevant Modes  RTSA

Parameters

<bool>  Persistence Infinite value:
0 or 1

Examples  DISP:VIEW:DENS:PERS:INF 1
          DISP:VIEW:DENS:PERS:INF 0

Query Syntax  DISP:VIEW:DENS:PERS?

Return Type  bool
Default  0

Last modified:
08-Aug-2016  Added for RTSA Mode (A.09.50)

DISPlay:WINDow:ANALog:LOWer <num>

(Read-Write) Set and query the Minimum scale value of the Power Meter display.

Relevant Modes  Power Meter, Pulse Measurements, CPM

Parameters

<num>  Minimum scale value. Choose a number between -100 and the upper (Max) scale value.

Examples  DISP:WIND:ANAL:LOW -70

Query Syntax  DISP:WIND:ANAL:LOWer?

Default  -100
DISPlay:WINDow:ANALog:UPPer <num>

(Read-Write) Set and query the Maximum scale value of the Power Meter display.

**Relevant Modes**
- Power Meter
- Pulse Measurements
- CPM

**Parameters**

- `<num>`: Maximum scale value. Choose a number between the upper (Max) scale value and 100.

**Examples**

DISP:WIND:ANAL:UPP -10

**Query Syntax**

DISPlay:WINDow:ANALog:UPPer?

**Default**

0

Last modified:

- 1-Apr-2014  Added CPM
- 31-Oct-2013  Added Pulse
**DISPlay:WINDow[:SELect] <int>**

*(Read-Write)* Set and query the selected trace display window. This command is used to modify existing pre-defined view, or create customized view. Window and Win State (*DISP:WIND:STAT*) keys are used to modify layout of view, the Data (*DISP:WIND:DATA*) command is used to specify the displayed data results for current window.

**Relevant Modes**  
IQA

**Parameters**

<integer>  
Choices: 1, 2, 3, or 4

**Examples**

DISP:WIND 3

**Query Syntax**

DISP:WIND?

**Return Type**  
Integer

**Default**  
1

---

Last Modified:

22oct2017  
Added IQA mode (10.1x)

---

**:DISPlay:WINDow:SPLit <char>**

*(Read-Write)* Set and query the multi-trace configuration.

Select a trace using *CALCulate:PARameter:SELect*

Change the measurement in each plot using *CALCulate:PARameter:DEFIne*

Change the format in each plot using *CALCulate SELected FORMat*

**Relevant Modes**  
NA and NF

**Parameters**

<char>  
Dual/Multi-trace configuration. Not case-sensitive. Choose from:

- **D1** - (x1)
- **D2** - (x2)
- **D12H** - (x2H)
- **D1123** - (x3H) - NA mode ONLY
- **D12_34** - (x4) - NA mode ONLY

**Examples**

DISP:WIND:SPL D2

**Query Syntax**

DISPlay:WINDow:SPLit?

**Default**  
D1
**DISPlay:WINDow:TRACe<n>:MEMory[:STATe] <bool>**

*(Read-Write)* Set and query the display state of a memory trace. A memory trace must already be stored using CALC:MATH:MEMorize.

To display both data and memory, set both to ON. **DISPlay:WINDow:TRACe<n>:STATe** sets the data trace ON.

**Relevant Modes**  CAT, NA, NF

**Parameters**

- `<n>` Trace number. If unspecified, value is set to 1.
  - For CAT and NF mode Choose from 1 to 2.
  - For NA mode Choose from 1 to 4.

- `<bool>` Choose from:
  - O or OFF - Memory trace NOT displayed
  - 1 or ON - Memory trace displayed

**Examples**

**DISPlay:WINDow:TRAC1:MEM 1**

**Query Syntax**  DISPlay:WINDow:TRACe<n>:MEMory:STATe?

**Default**  0 - Not displayed

---

**DISPlay:WINDow:TRACe<n>:STATe <bool>**

*(Read-Write)* Set and query the display state of the data trace. To display both data and memory, set both to ON. **DISPlay:WINDow:TRACe<n>:MEMory[:STATe]** sets the memory trace ON.

For NF mode, see also **DISP:WIND:TRAC<n>:STOR**.

**Relevant Modes**  CAT, NA, NF

**Parameters**

- `<n>` Trace number. If unspecified, value is set to 1.
  - For CAT mode Choose from 1 to 2.

---

Last Modified:

- 01-june-2018  Added Opt. 356 NF(10.3)
- 1-Nov-2012  removed CAT for (6.06)
- 26-Apr-2012  Modified for CAT (6.0)
For NA mode Choose from 1 to 4.

=bool> Choose from:

0 or OFF - Data trace NOT displayed
1 or ON - Data trace displayed

**Examples**

```
DISPlay:WINDow:TRAC1:STAT 1
```

**Query Syntax**

```
DISPlay:WINDow:TRACe2:STATe?
```

**Default**

1 - Displayed

---

**DISPlay:WINDow:TRACe<n>:STORe**

*(Write-Only)* Stores the current data trace into memory.

See also, `:DISP:WIND:TRAC<n>:STAT`.

**Relevant Modes** NF

**Parameters**

<n> Trace number for which display state is to be set or queried. Choose from 1 through 2.

**Examples**

```
DISP:WIND:TRAC:STOR
DISP:WIND2:TRAC:STOR
```

**Query Syntax** no query

**Return Type** Integer

**Default** <n> default = 1

---

**DISPlay:WINDow:ZOOM =bool>**

*(Read-Write)* Set and query the zoom window ON / OFF state.

**Relevant Modes** Pulse Measurements

**Parameters**

<bool> Choose from:
OFF or 0 - Zoom window OFF
ON or 1 - Zoom window ON

**Examples**
```
DISP:WIND:ZOOM 1
```

**Query Syntax**
```
DISP:WIND:ZOOM?
```

**Default**
OFF

---

**DISPlay:WINDow:TRACe:Y:DLiNe <num>**

*(Read-Write)* Set and query the display line Y-axis amplitude level. To turn ON/OFF display line, use **DISPlay:WINDow:TRACe:Y:DLiNe:STATe**.

**Relevant Modes**
IN, NF, SA, RTSA

**Parameters**

<numeric>  Y-axis amplitude level. Units depend on the selected setting. See [:SENSe]:AMPLitude:UNIT

**Examples**
```
DISP:WIND:TRAC:Y:DLIN -50
```

**Query Syntax**
```
DISP:WIND:TRAC:Y:DLIN?
```

**Return Type**
Numeric

**Default**
Depends on current units setting.

---

**DISPlay:WINDow:TRACe:Y:DLiNe:STATe <bool>**

*(Read-Write)* Set and query the ON/OFF state of the Display Line. Use **DISPlay:WINDow:TRACe:Y:DLiNe** to set the display level.

**Relevant Modes**
IQ, NF, SA, RTSA

**Parameters**

<boolean>  Choose from:

0 or OFF - Display Line OFF.
1 or ON - Display Line ON.
Examples  
**DisPlay:WINDow:TRACe:Y:DLIN:STAT 1**

**Query Syntax**  
DisPlay:WINDow:TRACe:Y:DLIne:STATe?

**Return Type**  
boolean

**Default**  
0 - OFF

Last Modified:
01June2018  Added NF mode Opt. 356 (10.3)
22Oct2017  Added IQA mode (10.1x)
10-June-2016  Added RTSA mode (9.50)
1-Aug-2011  New command

**DisPlay:WINDow:TRACe<n>:MEMory[:STATe] <bool>**

*(Read-Write)* Set and query the display state of a memory trace. A memory trace must already be stored using CALC:MATH:MEMorize.

To display both data and memory, set both to ON. **DisPlay:WINDow:TRACe<n>:STATe** sets the data trace ON.

**Relevant Modes**  
CAT, NA, NF

**Parameters**

<n>  
Trace number. If unspecified, value is set to 1.
For CAT and NF mode Choose from 1 to 2.
For NA mode Choose from 1 to 4.

<bool>  
Choose from:
O or OFF - Memory trace NOT displayed
1 or ON - Memory trace displayed

**Examples**  
**DisPlay:WINDow:TRACe1:MEM 1**

**Query Syntax**  
DisPlay:WINDow:TRACe<n>:MEMory:STATe?

**Default**  
0 - Not displayed

Last Modified:
01June2018  Added NF mode Opt. 356 (10.3)
26-Apr-2012  Modified for CAT (5.75)

**DisPlay:WINDow:TRACe<n>:STATe <bool>**

*(Read-Write)* Set and query the display state of the data trace. To display both data and memory, set both
to ON.  \texttt{DISPlay:WINDow:TRACe<n>:MEMory[STATE]} sets the memory trace ON.
For NF mode, see also \texttt{DISP:WIND:TRAC<n>:STOR}.

**Relevant Modes**  CAT, NA, NF

**Parameters**

\begin{itemize}
\item \textless{}\textit{n}\textgreater{}  Trace number. If unspecified, value is set to 1.
  For CAT mode Choose from 1 to 2.
  For NA mode Choose from 1 to 4.
\item \textless{}\textit{bool}\textgreater{}  Choose from:
  \texttt{O} or \texttt{OFF} - Data trace NOT displayed
  \texttt{1} or \texttt{ON} - Data trace displayed
\end{itemize}

**Examples**  \texttt{DISPlay:WINDow:TRAC1:STAT 1}

**Query Syntax**  \texttt{DISPlay:WINDow:TRACe2:STATe?}

**Default**  1 - Displayed

---

\texttt{DISPlay:WINDow:TRACe<n>:Y[:SCALe]:AUTO}  (Write-Only)  Autoscale the trace.

**Relevant Modes**  CAT, NA, NF, SA, FOPS

CAT mode: this command always acts on the selected trace. Select a trace using \texttt{CALCulate:PARameter:SELect}
SA mode: this command autoscales all displayed traces.

**Parameters**

\begin{itemize}
\item \textless{}\textit{n}\textgreater{}  For NA mode Only: Specify the displayed trace number. Choose from 1 to 4.
  For NF mode Only: Specify the displayed trace number. Choose from 1 to 2.
  If unspecified, value is set to 1.
\end{itemize}

**Examples**  \texttt{DISPlay:WINDow:TRAC1:Y:AUTO}

**Query Syntax**  Not Applicable

**Default**  Not Applicable

---

Last Modified:

- 01june2018  Added NF mode Opt. 356 (10.3)
- 26-Apr-2012  Modified for CAT (5.75)
- 01june2018  Added NF mode Opt. 356 (10.3)
- 19-Mar-2014  Added FOPS
**DISPlay:WINDow:TRACe<n>:Y[:SCALe]:BOTTom <num>**

(Read-Write) Set and query the bottom (lower) Y-axis scale value.

**Relevant Modes**  
CAT, NA, **NF**, SA

CAT mode: this command always acts on the selected trace. Select a trace using **CALCulate:PARameter:SELect**

SA - Read ONLY

**Parameters**

<n>  
NF and NA modes Only: Specify the displayed trace number.
NA Only: Choose from 1 to 4.
NF Only: Choose from 1 to 2.
If unspecified, value is set to 1.

<num>  
Bottom scale value. Units depend on the selected format.

**Examples**  
DISP:WIND:TRAC1:Y:BOTT -50

**Query Syntax**  
DISPlay:WINDow:TRAC<n>:Y[:SCALe]:BOTTom?

**Return Type**  
Numeric

**Default**  
Depends on Mode

---

**Last Modified:**

01June2018  
Added NF mode Opt. 356 (10.3)

23-Jan-2013  
Modified again for CAT

26-Apr-2012  
Modified for CAT (5.75)

**DISPlay:WINDow:TRACe<n>:Y[:SCALe]:PDIVision <num>**

(Read-Write) Set and query the Y-axis scale per division value. Use this command with **DISPlay:WINDow:TRACe:Y[:SCALe]:RPOsition** and **DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel** which establish the reference position and level.

**Relevant Modes**  
CAT, **NF**, NA, SA, RTSA, **FOPS**

CAT mode: This command always acts on the selected trace. Select a trace using **CALCulate:PARameter:SELect**

SA mode: Applicable ONLY when [:SENSe]:AMPLitude:SCALe = LOG; (Units are always "dB")
Parameters
<n>  NA mode Only: Specify the displayed trace number. Choose from 1 to 4.
     For NF mode Only: Specify the displayed trace number. Choose from 1 to 2.
     If unspecified, value is set to 1.
<num> Scale value. Units depend on the selected format.

Examples  DISP:WIND:TRAC1:Y:PDIV -.25

Query Syntax  DISP:WIND:TRAC<n>:Y[:SCALe]:PDIVision?

Return Type  Numeric

Default  Depends on Mode

Last Modified:
01 June 2018  Added NF mode Opt. 356 (10.3)
10 June 2016  Added RTSA mode (9.50)
19 Mar 2014  Added FOPS
11 Nov 2013  Modified again
26 Apr 2012  Modified for CAT (5.75)

**DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RLEVel <num>**

*(Read-Write)* Set and query the Y-axis value of the reference line. Use this command with **DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RPOSition** which sets the position of the reference.

**Relevant Modes**  CAT, NA, NF, SA, RTSA, FOPS

CAT mode: This command always acts on the selected trace. Select a trace using **CALCulate:PARameter:SELect**

SA mode: This command acts on all displayed traces.

**Parameters**
<n>  NA mode Only: Specify the displayed trace number. Choose from 1 to 4.
     For NF mode Only: Specify the displayed trace number. Choose from 1 to 2.
     If unspecified, value is set to 1.
<num>  Y-axis reference level. Units depend on the selected format.

**Examples**  DISP:WIND:TRAC1:Y:RLEV -50

**Query Syntax**  DISP:WIND:TRAC<n>:Y[:SCALe]:RLEVel?

**Return Type**  Numeric

**Default**  Depends on Mode
DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RPOSition <num>

(Read-Write) Set and query the position of the Y-axis reference line. Use this command with RLEVel which sets the Y-axis value of the reference line.

**Relevant Modes**
- CAT, NA, NF, SA, RTSA, FOPS
  - CAT mode: this command always acts on the selected trace.
  - SA mode: this command acts on all displayed traces.

**Parameters**
- `<n>` NA mode Only: Specify the displayed trace number. Choose from 1 to 4.
  - For NF mode Only: Specify the displayed trace number. Choose from 1 to 2.
  - If unspecified, value is set to 1.
- `<num>` Reference position. Choose a value from 0 to 10.

**Examples**
```
DISP:WIN:TRAC1:Y:RPOS -50
```

**Query Syntax**
```
DISPlay:WINDow:TRACe<n>:Y[:SCALe]:RPOSition?
```

**Return Type** Numeric

**Default** Depends on Mode

---

Last Modified:
- 01-june-2018 Added NF mode Opt. 356 (10.3)
- 10-june-2016 Added RTSA mode (9.50)
- 20-Mar-2014 Added FOPS
- 23-Jan-2013 Modified again
- 26-Apr-2012 Modified for CAT (5.75)
Relevant Modes

CAT, NA, NF, SA
CAT mode: this command always acts on the selected trace.
SA mode: Read ONLY

Parameters

<n> NA and NF mode Only: Specify the displayed trace number.
NA: Choose from 1 to 4.
NF: Choose from 1 to 2.
If unspecified, value is set to 1.

<num> Top scale value. Units depend on the selected format.

Examples

DISP:WIND:TRAC1:Y:TOP 0

Query Syntax

DISPlay:WINDow:TRACe<n>:Y[:SCALe]:TOP?

Return Type

Numeric

Default

Depends on Mode

Last Modified:

01-june-2018 Added Opt. 356 NF(10.3)
23-Jan-2013 Modified again
26-Apr-2012 Modified for CAT (5.75)

DISPlay:WINDow<n>:DATA <char>

(Read-Write) Set and query the displayed data results for the selected trace display window. The Data command is used to specify the displayed data results for current window.
See also Window (DISP:WIND) and Win State (DISP:WIND:STAT) commands.

Relevant Modes

IQA

Parameters

<n> Trace number for which display state is to be set or queried. Choose from 1 through 4.

<character> Spectrum Measurement - two data type results are available:
- SPECtrum (Magnitude Spectrum)
- TIMesummary (Time Summary)
For Waveform Measurements - six data types are available:
- REAL (I vs. Time)
- IMAG (Q vs. Time)
- POLar (Q vs. I)
- WAVeform (RF Envelope)
- PHASe (Phase vs. Time)
- UPHase (Unwrapped Phase vs. Time)
- TIMesummary (Time Summary) Time Summary result includes the capture time, spectrum Analysis time, and waveform Start/Stop time.

Examples
- `DISP:WIND3:DATA UPH`
- `DISP:WIND4:DATA SPEC`

Query Syntax
- `DISP:WIND<n>:DATA?`

Return Type
- Character

Default
- Spectrum: SPEC|SPEC|SPEC|SPEC
- Waveform: WAV|WAV|WAV|WAV
- User Defined: SPEC|WAV|REAL|IMAG

Last Modified:
- 22oct2017 Added IQA mode (10.1x)

**DISPlay:WINDow<n>:STATe <int>**

(Read-Write) Set and query to enable or disable the selected trace display window. This command is used to modify existing pre-defined view, or create customized view. Window (`DISP:WIND`) and Win State commands are used to modify layout of view, the Data (`DISP:WIND:DATA`) command is used to specify the displayed data results for current window.

Note: Initial User Defined default displays all data results from Spectrum measurement and Waveform measurement are available (i.e., all four displays).

Relevant Modes
- IQA

Parameters
- `<n>` Trace number for which display state is to be set or queried. Choose from 1 through 4.
- `<integer>` Choices: 1, 2, 3, or 4

Examples
- `DISP:WIND3:STAT 1`
- `DISP:WIND4:STAT 0`

Query Syntax
- `DISP:WIND<n>:STAT?`

Return Type
- Integer

Default
- Spectrum | Waveform: ON|OFF|OFF|OFF
- User Defined: ON|ON|ON|ON

Last Modified:
- 22oct2017 Added IQA mode (10.1x)
**FORMat:BORDer <char>**

*(Read-Write)* Set the byte order used for data transfer. This command is only implemented if `FORMAT:DATA` is set to :REAL. If `FORMAT:DATA` is set to :ASCII, the swapped command is ignored.

**Relevant Modes**  ALL

**Parameters**

<char> Choose from:

- **NORMAL** - LSB first; for "x86 or "Little Endian" computers .
- **SWAPPed** - MSB first. Use when your controller is anything other than an IBM compatible computers. Use **SWAPPed** if you are using VEE, LabView, or T&M Tool kit.

**NOTE:**
These descriptions may be reversed from other Keysight Instruments, but compliant with IEE 488.2.

**Examples**

- FORM:BORD NORM
  
  See example parsing program in C#.

**Query Syntax**  FORMat:BORDer?

**Default**  NORMAL

Last Modified:

31-Oct-2013  New command

---

**FORMat[:DATA] <char>**

*(Read-Write)* Set and query the format (binary or ASCII) by which FieldFox data is read.

Data is read using the following commands:

- **NF**, SA Mode: `TRACe:DATA?`
- Calibration data (NA, CAT, and VVM modes): `[:SENSe]:CORRection:COEFFicient[:DATA]`
- To change the byte order, refer to :FORM:BORD.

**Note:** Do NOT do Binary Block transfers (REAL, 32 or REAL, 64) when using over Telnet to port 5024 on FieldFox.

**Relevant Modes**  ALL

**Parameters**
Choose from:

- **REAL,32** - Best for transferring large amounts of measurement data. (Binary data)
- **REAL,64** - Slower but has more significant digits than REAL,32. Use REAL,64 if you have a computer that doesn't support REAL,32. (Binary data)
- **ASCii,0** - The easiest to implement, but very slow. Use when you have small amounts of data to transfer. ASCii,0 returns a carriage return ("\n") terminated, comma-separated list of numbers, which might be re,im pairs, or could be scalar numbers corresponding to the current instrument format (such as LogMag) for FDATA?

**Notes:**

- The REAL,32 and REAL,64 arguments transfer data in block format. The byte order is *little endian*. FORM:BORDer, which reverses the byte order, is NOT supported.
- If the I/Q data format is set to ASCii,0, the maximum I/Q data length is 50k.

**Block Data**

The following graphic shows the syntax for definite block data.

Only the **Data Byte** is sent in either Binary or ASCii format.

All other characters are ASCii format.

- `<num_digits>` specifies how many digits are contained in `<byte_count>`
- `<byte_count>` specifies how many data bytes will follow in `<data bytes>`

**Example of Definite Block Data**

```
#17ABC+XYZ
```

Where:

- `#` - always sent before definite block data (ascii format)
- `1` - specifies that the byte count is one digit (7) (ascii format)
- `7` - specifies the number of data bytes that will follow, not counting `<NL>`<END> (ascii format)
- `ABC+XYZ` – Data (binary or ascii format)
- `<NL>` - always sent at the end of block data to indicate the last character being sent (program message terminator) - (ascii format)
- `<END>` - just for illustrative purposes and indicates the end of the data transmission.

**Binary Data Notes:**

The format is described by either REAL,32 (which indicates that each number takes 4 bytes) or REAL,64 (which indicates that each number takes 8 bytes). No separator is necessary between the numbers.

For a trace with 1001 complex points (real, imag) in REAL,64 format, the
header would describe a 1001*2*8=16016 byte block, so it would be: #516016 followed by 16016 bytes of data.

For a trace with 5 data points in LogMag format in REAL,32 format, the header would describe a 5*1*4=20 byte block, so it would be: #220 followed by 20 bytes of data.

**Examples**

FORM REAL,32

See example parsing program in C#.

**Query Syntax**

FORMat[:DATA]?

**Default**

ASCii,0

---

**INITiate:IQC**

*(Write-Only)* Causes the FieldFox to start the I/Q capture process. When in Single sweep (acquisition for I/Q) mode, the IQA performs the current defined I/Q capture before going to the hold state again.

This command is overlapped. This means that it will execute and, while the FieldFox performs a single sweep (acquisition for I/Q), it will also accept new commands. Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands.

**Note:** This command can only be used in single acquisition mode. It is ignored when the I/Q capture is set to continuous acquisition. To enable/disable continuous acquisition, refer to INITiate:CONTinuous.

**Relevant Modes**

IQA

**Examples**

INIT:IQC;*OPC?

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

**INITiate:CONTinuous <bool>**

*(Read-Write)* Specify acquisition mode for the FieldFox.

The OFF setting for this command is overlapped. This means that it will execute and also accept new commands while switching to single sweep (acquisition for I/Q) mode.
Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands. See also, INITiate:IQCapture.

**Relevant Modes** ALL

**Parameters**

<boolean> Acquisition mode. Choose from:

0 or OFF - Single acquisition mode. Use INIT:IMM to trigger a acquisition.
1 or ON - Continuous acquisition mode

**Examples** INIT:CONT 0;*OPC?

See example in VVM mode

**Query Syntax** INITiate:CONTinuous?

**Return Type** Boolean

**Default** ON - Continuous

Last Modified:

22oct2017 Added IQA mode (10.1x)

---

**INITiate[:IMMediate]**

*(Write-Only)* Causes the FieldFox to perform a single sweep, then hold. Use this sweep mode for reading trace data. First send INIT:CONT 0 to set single sweep mode.

When in Continuous sweep mode, this command is ignored.

This command is overlapped. This means that it will execute and, while the FieldFox performs a single sweep, it will also accept new commands. Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands.

**Relevant Modes** ALL

**Examples** INIT;*OPC?

**Query Syntax** Not Applicable

**Default** Not Applicable

---

**INITiate:RESTart**

*(Write-Only)* Behavior is mode and Sweep type (Continuous/Hold/Single) dependent.

**SA/RTSA/IQA:**

Causes the FieldFox to restart trace averaging at count of 1. When in Single sweep (acquisition for I/Q) mode, the IQ/RTSA/SA performs a complete set of averages before going to the hold state again.

This command is overlapped. This means that it will execute and, while the FieldFox performs a single sweep (acquisition for I/Q), it will also accept new commands. Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands.

In Single sweep (acquisition for I/Q) mode, the IQA/RTSA/SA performs a complete set of averages before
going to the hold state again.

**NA and CAT:**

In Hold mode:

1) Causes the FieldFox to restart trace averaging at count of 1.
2) Takes a single sweep.

**Note:** Does not support the SA/IQA/RTSA behavior of acquiring ALL averages when average mode is set to SWEEP average mode. If in SWEEP average mode, will only completes the first of the required averages. `INIT:IMM` command is required to keep adding to the average trace until desired average count has been reached.

In Continuous sweep mode:

1) Resets the average counter and continue free run sweeping. Similar to Hold mode, if in sweep average mode, will start acquiring sweeps to collect the required averages and will continue sweeping. Average counter will climb to set value (This set value displays on the Fieldfox).
2) If in point average mode, will acquire all averages in a single sweep as expected, and will continue acquiring data.

**Note:** If you are in Continuous sweep mode, resets averaging at a count of 1.

### Relevant Modes

CAT, IQA, LTE FDD, 5GTF, NA, SA, RTSA

### Examples

```
INIT:REST;*OPC?
```

### Query Syntax

Not Applicable

### Default

Not Applicable

---

**:INPut:LAN:IDENtify:SNUMber <number>**

(Read-Write) Specify the last 5 digits for a target power sensor serial number to auto-generate the hostname.

**Relevant Modes**

Power Meter (USB Sensor), Pulse Modes: Supports Options 208 Frequency Offset Mode, 310 Power Meter Mode (USB), and 330 Pulse Measurements

**Parameters**

```
<number> 12345
```

**Examples**

```
:INP:LAN:IDEN:SNUM 12345
```

**Query Syntax**

```
:INP:LAN:IDEN:SNUM?
```

**Return Type**

---

Last Modified:

- **27nov2018** Added CAT, LTE FDD, & 5GTF modes and updated descriptions (11.0)
- **22oct2017** Added IQA mode (10.1x)
- **10-june-2016** Added RTSA mode (9.50)
**:INPut:LAN:IDENTify:TYPe**  SNUM | IPA

*(Read-Write)* Specifies where to find a LAN hostname.

- **Relevant Modes**: Power Meter (USB Sensor), Pulse Modes: Supports Options 208 Frequency Offset Mode, 310 Power Meter Mode (USB), and 330 Pulse Measurements

- **Parameters**
  - `<char>` **SNUM** - use autofind (i.e., default uses the device serial number)
  - **IPA** - user enters an IP address.

- **Examples**
  - `:INP:LAN:IDENT:TYP IP 168.212.226.204`

- **Query Syntax**: `:INP:LAN:IDENT:TYP?`

- **Return Type**: Numeric

- **Default**: SNUM

---

**INSTrument:CATalog?**

*(Read-Only)* Read the modes available on the FieldFox.

- **Relevant Modes**: ALL

- **Examples**
  - `Inst:Cat?`
  - **Returns**: NA,SA,Power Meter,CAT

- **Return Type**: Comma-separated list of strings.

- **Default**: Not Applicable

**INSTrument:GTL**

*(Write-Only)* Returns front panel key press control. To lockout front panel operation, send `INST:GTR`.

- **Relevant Modes**: ALL
Examples

INST:GTR

Query Syntax
Not Applicable

Default
Not Applicable

Last Modified:
20-Nov-2014 New command (7.75)

INSTrument:GTR

(Write-Only) Locks out the front panel key presses. RMT is displayed on the screen. To return to front panel operation, press ESC or send INST:GTL.

Relevant Modes
ALL

Examples
INST:GTR

Query Syntax
Not Applicable

Default
Not Applicable

Last Modified:
20-Nov-2014 New command (7.75)

INSTrument[:SELect] <string>

(Read-Write) Set and query the current operating mode. Use Inst:CAT? to read the available modes on the FieldFox.

This command is overlapped. This means that it will execute and, while the FieldFox changes mode, it will also accept new commands. Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands.

Relevant Modes
ALL

Parameters

<string> Operating Mode. Case-sensitive. Choose from the modes that are installed on your FieldFox:

- "CAT"
- "IQ"
- "NA"
- "SA"
- "Power Meter"
- "VVM"
- "Pulse Measurements"
- "ERTA"
Examples

**INST "NA";"OPC?**

Query Syntax

**INSTRument[:SELect]?**

Default

Depends on FieldFox model

---

Last Modified:

05Oct2017  Added IQA mode (10.1x)

---

### :LTEFdd:DATA:GPS?

*(Read-Only)* Returns LTE FDD result with GPS information.

**Relevant Modes**  LTE FDD

**Parameters**  None

**Examples**  **LTEF:DATA:GPS?**

**Default**  Not Applicable

---

Last Modified:

19-Dec-2018  New command

---

### :LTEFdd:DATA?

*(Read-Only)* Returns LTE FDD result without GPS information.

**Relevant Modes**  LTE FDD

**Parameters**  None

**Examples**  **LTEF:DATA?**

**Default**  Not Applicable

---

Last Modified:

19-Dec-2018  New command

---

### MMEMory:CATalog? [string]

*(Read-Only)* Returns a comma-separated string of file names.

See ALL MMEM commands.
### MMEMory:CDIRectory <string>

(Read-Write) Change the active drive/folder. All subsequent MMEM commands will use this drive/folder as the default.

**Relevant Modes**

| ALL |

**Parameters**

<table>
<thead>
<tr>
<th>&lt;string&gt;</th>
<th>Folder name enclosed in quotes. Case-sensitive. Include brackets, and colon. Specify the following drives:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;[INTERNAL]:&quot;</td>
<td>FieldFox internal drive.</td>
</tr>
<tr>
<td>&quot;[SDCARD]:&quot;</td>
<td>External SD card.</td>
</tr>
<tr>
<td>&quot;[USBDISK]:&quot; or &quot;[USBDISK1]:&quot;</td>
<td>The first USB flash drive to be plugged into either of the FieldFox USB slots.</td>
</tr>
<tr>
<td>&quot;[USBDISK2]:&quot;</td>
<td>The second USB flash drive to be plugged into either of the FieldFox USB slots.</td>
</tr>
</tbody>
</table>

Specify a folder on the drive by appending the folder name within the quotes. See examples.

The following arguments are superseded (but still accepted):

- An SD card is specified as "\Storage Card"
- A USB flash drive is specified as "\Hard Disk"
- The internal memory is specified as "\UserData"
**Examples**

- Change active drive to internal
  
  `MMEM:CDIR "[INTERNAL]:"`

- On the internal drive, specify the Keywords folder
  
  `MMEM:CDIR "[INTERNAL]:\Keywords"`

- Change active drive to the USB flash disk
  
  `mmemory:cdirectory "[USBDISK]:"`

**Query Syntax**

- `MMEMory:CDIRectory?`

  Folders on the active drive are shown in brackets as `[My Folder]`

**Return Type**

- String

**Default**

- `"[INTERNAL]:"`

---

**MMEMory:COPY <file1>,<file2>**

*(Write-Only)* Copies file1 to file2. Extensions must be specified.

See **MMEM:CDIR** to learn how to set the active drive and how to specify a drive/folder.

See ALL MMEM commands.

**Relevant Modes**

- ALL

**Parameters**

- `<file1>` String - Drive, folder, filename, and extension of the file to be copied. If drive and folder are unspecified, the active drive/folder is used.

- `<file2>` String - Drive, folder, filename, and extension to be created from `<file1>`. If unspecified, the active drive/folder is used.

**Examples**

- Copies the existing file to the same folder with a new file name.
  
  `MMEM:COpy "MyFile.sta", "MyFile.sta"`

- Copies the existing file from the active drive to the USB drive with the same file name.
  
  `MMEM:COpy "MyFile.sta", "[USBDISK]\MyFile.sta"`

**Query Syntax**

- Not Applicable

**Default**

- Not Applicable

---

**Last Modified:**

- 23-Apr-2012 Major modifications
**MMEMory:DATA <filename>,<data>**

*(Read-Write)* Read and store `<data>` into the file `<filename>`. Because this is block data, any file format can be read.

See [MMEM:CDIR](#) to learn how to set the active drive and how to specify a drive/folder. See [ALL MMEM commands](#) and [Memory Command Examples](#).

**Relevant Modes**  
ALL

**Parameters**

- `<filename>`  
  String - Drive, folder, filename, and extension into which data will be loaded. If drive and folder are unspecified, the active drive/folder is used.

- `<data>`  
  Data in 488.2 block format.

**Examples**

- *Stores data into "MyFile"*
  
  MMEM:DATA "MyFile",<data>

- *Stores data into "MyFile" on USB*
  
  MMEM:DATA "[USBDISK]\MyFile",<data>
  
  [See a transfer image to PC example using this command](#).

**Query Syntax**  
MMEMory:DATA? <filename>

**Default**  
Not Applicable

---

**MMEMory:DELete <string>**

*(Write-Only)* Deletes the specified file.

See [MMEM:CDIR](#) to learn how to set the active drive and how to specify a drive/folder. See [ALL MMEM commands](#).

**Relevant Modes**  
ALL

**Parameters**

- `<string>`  
  String - Drive, folder, filename, and extension to delete. If unspecified, the active drive/folder is used.

**Examples**

- *Delete file from active drive/folder*
  
  MMEM:DEL "MyOldFile.sta"

- *Delete file from USB*
  
  MMEM:DEL "[USBDISK]\MyOldFile.sta"

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

Last Modified:

- 17-Dec-2013  
  Added link to example

- 23-Apr-2012  
  Major modifications
**MMEMory:IMPort:DUT  <string>,<char>**

(Write-only) Load a match data (*.csv or *.snp) file from the specified device.
- Imports *.csv and *.snp files from the "NoiseFigure\DUT" folder.

Note: Before importing a file, the file must be stored in the NoiseFigure\DUT folder.
See also, **MMEMory:STORE:DUT** and **MMEMory:LOAD:DUT**.
See ALL MMEM commands.

Relevant Modes  **NF**

Parameters
- **<string>**  *.csv or *.snp
- **<char>**  Memory device where the match data file is stored. Choose from the following:
  - **INTernal** - internal storage device
  - **USB** - first USB device
  - **SD** - SD storage card

Examples  
**MMEM:IMP:DUT "MySNP.s2p",INT**

Query Syntax  Not Applicable

Default  Not Applicable

---

Last Modified: 01june2018  Added NF mode Opt. 356 (10.3)

---

**MMEMory:IMPort:PAMPlifier  <string>,<char>**

(Write-only) Load a match data (*.csv or *.snp) file from the specified device.
- Imports *.csv and *.snp files from the "NoiseFigure\Preamp" folder.

Note: Before importing a file, the file must be stored in the NoiseFigure\Preamp folder.
See also, **MMEMory:LOAD:PAMPlifier** and **MMEMory:STORE:PAMPlifier**.
See ALL MMEM commands.

Relevant Modes  **NF**

Parameters
- **<string>**  *.csv or *.snp
- **<char>**  Memory device where the match data file is stored. Choose from the following:
  - **INTernal** - internal storage device
MMEMory:LOAD:PAMPifier <string>,<char>

(Write-only) Load an uncertainty preamplifier (*.amp) file from the specified device.

- Preamplifier files (*.amp) are saved and recalled from the "NoiseFigure\AMP" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

Note: Template *.amp files for several common models of preamplifiers may be found in the default AMP table directory of the FieldFox. In these files, the reflection coefficients have been set to the values specified for the respective model numbers, but the values for the preamplifier or preamplifier uncertainty will need to be entered at each frequency.

See also, MMEMory:STORe:PAMPifier and MMEMory:IMPorT:PAMPifier.

See ALL MMEM commands.

Relevant Modes NF

Parameters

<string> For *.amp files, provide filename of the *.amp file WITHOUT an extension.

<char> Memory device where the preamplifier file (*.amp) is stored. Choose from the following:

- INTernal - internal storage device
- USB - first USB device
- SD - SD storage card

Examples MMEM:LOAD:PAMP "MyAMP", INT

Query Syntax Not Applicable

Default Not Applicable

Last Modified:
01june2018 Added NF mode Opt. 356 (10.3)
command.

For SA (Only), to load a source antenna file, use MMEMory:LOAD:SANTenna.

- All Channel Scanner (ChScn) and SA mode Antenna files are saved and recalled as *.csv files, which allows them to also be read by spreadsheet programs.
- The FieldFox can also read *.ANT (Antenna) files that were created from older Keysight Spectrum Analyzers.
- Antenna files are saved and recalled from the "Antenna" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

See ALL MMEM commands.

Relevant Modes  
ChScn, SA

Parameters

<string>  For *.csv files, provide filename of the antenna file WITHOUT an extension.  
For *.ant files, provide filename and .ant extension.

<char>  Memory device where the Antenna file is stored. Choose from the following:
  - INTernal - internal storage device
  - USB - first USB device
  - SD - SD storage card

Examples  
MMEM:LOAD:ANT "MyAntenna", INT
mmemory:load:antenna "demo_antenna_26m_3g.ant",internal

Query Syntax  Not Applicable

Default  Not Applicable

Last Modified:
01June2018  Added Channel Scanner (ChScn) - (10.3)
26-Jan-2015  Modified for source
16-Dec-2014  Fixed .ant recall
29-Oct-2010  New command (5.30)

MMEMory:LOAD:CABLe <string>,<char>

(Read-Write) Loads a receiver cable file from the specified device. If using only one cable file, use this command.

To load a source cable file, use MMEMory:LOAD:SCABle

- Channel Scanner (ChnScn) and SA Cable files are saved as *.csv files, which allows them to also be read by spreadsheet programs.
- DTF Cable files are saved as *.xml files.
- Cable files are saved and recalled from the "Cables" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.
See ALL MMEM commands.

Relevant Modes  

**ChScn**, **CAT**, **SA**

Parameters

<string> SA mode: Filename of the cable file without an extension. 
CAT mode: Filename of the cable file with an extension.

<char> Memory device where the cable file is stored. Choose from:
- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

Examples

- **MMEM:LOAD:CABL "MyCable",INT */ChnScn and SA modes**
- **MMEM:LOAD:CABL "MyCable.xml",INT */CAT mode**

Query Syntax  Not Applicable

Default  Not Applicable

Last Modified:

27 nov 2018  Updated filename load descriptions with mode specific content.
01 June 2018  Added Channel Scanner (ChScn) - (10.3)
26 Jan 2015  Modified for source
29 Oct 2010  New command (5.30)

**MMEMory:LOAD:DUT <string>,<char>**

(Write-only) Load a uncertainty DUT (*.dut) file from the specified device.

- DUT files are saved and recalled from the "NoiseFigure\DUT" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

See also, **MMEMory:STORE:DUT** and **MMEMory:IMPort:DUT**.
See ALL MMEM commands.

Relevant Modes  **NF**

Parameters

<string> For *.dut files, provide filename of the *.dut file without an extension.

<char> Memory device where the DUT file is stored. Choose from the following:
- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

Examples  **MMEM:LOAD:DUT "MyDUT",INT**

Query Syntax  Not Applicable
**Default**  Not Applicable

---

**Last Modified:**

01june2018  Added NF mode Opt. 356 (10.3)

---

**MMEMory:LOAD:ENR <string>,<char>**

*(Write-only)* Load a noise figure ENR (*.enr) file from the specified device.

- ENR files are saved and recalled from the "NoiseFigure\ENR" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

Note: Template *.enr files for several common models of noise sources may be found in the default ENR table directory of the FieldFox. In these files, the reflection coefficients have been set to the values specified for the respective model numbers, but the values for the ENR or ENR uncertainty will need to be entered at each frequency.

See also, **MMEMory:STORE:ENR**.

See ALL MMEM commands.

**Relevant Modes**  **NF**

**Parameters**

- `<string>`  For *.enr files, provide filename of the *.enr file WITHOUT an extension.
- `<char>`  Memory device where the ENR file is stored. Choose from the following:
  - **INTernal** - internal storage device
  - **USB** - first USB device
  - **SD** - SD storage card

**Examples**  **MMEM:LOAD:ENR "MyENR", INT**

**Query Syntax**  Not Applicable

**Default**  Not Applicable

---

**Last Modified:**

01june2018  Added NF mode Opt. 356 (10.3)

---

**:MMEMory:LOAD:LIST**

*(Write-only)* Load a *.csv custom file.

See ALL MMEM commands.

**Relevant Modes**  **ChScn**

**Parameters**

- `<string>`  For *.csv files, provide filename of the antenna file WITHOUT an extension.
Memory device where the Antenna file is stored. Choose from the following:

- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

**Examples**

```
MMEM:LOAD:ANT "ChanScan",INT
mmemory:load:antenna "demo_channel_3g.csv",internal
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

**Last Modified:**

19-Dec-2018 New command

---

### :MMEMory:LOAD:OTA:LOG <string>

*(Write-only)* Load a *.csv log file.

See ALL MMEM commands.

**Relevant Modes**

**LTE FDD, 5GTF**

**Parameters**

- `<string>` For *.csv files, provide filename and .csv.

**Examples**

```
MMEM:LOAD:OTA:LOG "MyOTAfile.csv"
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

**Last Modified:**

19-Dec-2018 New command

---

### :MMEMory:LOAD:LOG

*(Write-only)* Load a *.csv or *.kml log file.

See ALL MMEM commands.

**Relevant Modes**

**ChScn**

**Parameters**

- `<string>` For *.csv files, provide filename and .csv. For *.kml files, provide filename and .kml extension.

**Examples**

```
MMEM:LOAD:LOG
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

Memory device where the Antenna file is stored. Choose from the following:
• **INTernal** - internal storage device
• **USB** - first USB device
• **SD** - SD storage card

### Examples

- `MMEM:LOAD:ANT "MyAntenna",INT`
- `mmemory:load:antenna "demo_antenna_26m_3g.kml",internal`

### Query Syntax
Not Applicable

### Default
Not Applicable

---

**MMEMory:LOAD:STATe <string>**

*(Write-Only)* Loads an instrument state file. This command CAN be used with *OPC?*.

**Relevant Modes** ALL

**Parameters**

- `<string>` Filename and extension of the state file.

**Examples**

- `MMEM:LOAD:STATe "AutoSave1.sta"`

**Query Syntax** Not Applicable

**Default** Not Applicable

---

**Last Modified:**

- 11-Nov-2013 Linked to Q
- 28-Mar-2013 Removed OPC? limitation
- 26-Jul-2012 Added note about *OPC?*

---

**MMEMory:MDIRectory <string>**

*(Write-Only)* Makes a new folder. See **MMEM:CDIR** to learn how to set the active drive/folder and how to specify a drive/folder. See ALL MMEM commands.

**Relevant Modes** ALL

**Parameters**
<string> Drive/folder to create. If unspecified, the folder is created in the active drive/folder.

Examples
- 'Folder is created in the active drive/folder
  MMEMory:MDIRectory "MyFolder"
- 'Folder is created on the USB drive.
  MMEMory:MDIRectory "[USBDISK]\MyFolder"

Query Syntax Not Applicable
Default Not Applicable

Last Modified:
23-Apr-2012 Major modifications

MMEMory:MOVE <file1,<file2>

(Write-Only) Renames <file1> to <file2>. File extensions must be specified.
See MMEM:CDIR to learn how to set the active drive/folder and how to specify a drive/folder.
See ALL MMEM commands.

Relevant Modes ALL

Parameters
- <file1> String - Drive, folder, filename, and extension of the file to be renamed. If unspecified, the active drive/folder is used.
- <file2> String - Drive, folder, filename, and extension of the new file. If unspecified, the active drive/folder is used.

Examples
- 'Rename file from/to active drive/folder.
  MMEM:MOVE "OldFile.sta", "NewFile.sta"
- 'Rename file from active drive/folder to USB drive.
  MMEM:MOVE "OldFile.sta", "[USBDISK]\NewFile.sta"

Query Syntax Not Applicable
Default Not Applicable

Last Modified:
23-Apr-2012 Major modifications

MMEMory:RDIRectory <string>[,<recursive>]

(Write-Only) Removes the specified folder if it is empty. If the folder is NOT empty, a 'Media Protected' message appears.
See MMEM:CDIR to learn how to set the active drive/folder and how to specify a drive/folder.
See ALL MMEM commands.
Relevant Modes  
ALL

Parameters

<string>  
String - Drive, folder from which folder is to be removed. If unspecified, the active drive/folder is used.

<recursive>  
String - Optional. Case-sensitive.  
"recursive" - Removes everything in the specified folder and all sub-folders and their contents.

Examples  
'Removes 'oldfolder' from active drive if it is empty.  
MMEMory:RDIR "OldFolder"
'Removes 'oldfolder' from USB drive  
MMEMory:RDIR "[USBDISK]:\OldFolder"
'Removes 'oldfolder' and all sub-folders  
MMEMory:RDIR "OldFolder","recursive"

Query Syntax  
Not Applicable

Default  
Not Applicable

Last Modified:
1-Apr-2014  Added optional argument
23-Apr-2012  Major modifications

:MMEMory:STATe:STORe:LIST

(Write-only) Save a csv custom list file.  
See ALL MMEM commands.

Relevant Modes  
ChScn

Parameters

<string>  
For *.kml files, provide filename and .kml extension.

<char>  
Memory device where the Antenna file is stored. Choose from the following:

- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

Examples  
MMEM:LOAD:ANT "ChanScan",INT

Query Syntax  
Not Applicable

Default  
Not Applicable

Last Modified:
MMEMory:STORe:ANTenna <string>,<char>

(Write-only) Saves the currently-loaded receiver antenna table to a file on the specified device.

- All Channel Scanner (ChScn) and SA modes the Antenna files are saved and recalled as *.csv files, which allows them to also be read by spreadsheet programs.
- The FieldFox can also read, but NOT save, *.ANT (Antenna) files. These files were created from older Keysight Spectrum Analyzers.
- Antenna files are saved and recalled from the "Antenna" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

See ALL MMEM commands.

Relevant Modes ChScn, SA

Parameters

<string> Filename of the antenna file WITHOUT an extension.
<char> Memory device where the antenna file is to be stored. Choose from:
- INTernal - internal storage device
- USB - first USB device
- SD - SD storage card

Examples MMEM:STOR:ANT "MyAntenna_1",USB

Query Syntax Not Applicable

Default Not Applicable

Last Modified:

01June2018 Added Channel Scanner (ChScn) - (10.3)
16-Dec-2014 Edit to *.ant files
22-Aug-2012 Fixed example
22-Oct-2010 New command (5.30)

MMEMory:STORe:CABLe <string>,<char>

(Read-Write) Saves the currently-loaded receiver cable table to a file at the specified location and device. Channel Scanner (ChScn) and SA Cable files are saved as *.csv files, which allows them to also be read by spreadsheet programs.
DTF Cable files are saved as *.xml files.
Cable files are saved and recalled from the "Cables" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

See ALL MMEM commands.
**Relevant Modes**  
CAT (DTF), ChScn, SA

**Parameters**

<string>  
ChScn and SA mode: Filename of the cable file without an extension.  
CAT mode: Filename of the cable file with an extension.

<char>  
Memory device where the file is to be stored. Choose from:
- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMEM:STOR:CABL &quot;MyCable&quot;,INT</td>
<td>*/ChScn and SA mode</td>
</tr>
<tr>
<td>MMEM:STOR:CABL &quot;MyCable.xml&quot;,INT</td>
<td>*/CAT (DTF) mode</td>
</tr>
</tbody>
</table>

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

**Last Modified:**

27 Nov 2018  
Updated filename store with mode specific content.

01 June 2018  
Added Channel Scanner (ChScn) - (10.3)

26 Jan 2015  
Modified for source

22 Aug 2012  
Fixed example

22 Oct 2010  
New command (5.30)

---

**MMEMory:STOR:e:DUT**  
<string>,<char>

*(Write-Only)*  
Store a uncertainty DUT (*.dut) file to the specified device.

- ENR files are saved and recalled from the "NoiseFigure\DUT" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

See also, **MMEMory:LOAD:DUT** and **MMEMory:IMPort:DUT**.  
See ALL MMEM commands.

**Relevant Modes**  
NF

**Parameters**

<string>  
For *.dut files, provide filename of the *.dut file without an extension.

<char>  
Memory device where the DUT file is stored. Choose from the following:
- **INTernal** - internal storage device
- **USB** - first USB device
- **SD** - SD storage card

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMEM:STOR:DUT &quot;MyDUT&quot;,INT</td>
<td></td>
</tr>
</tbody>
</table>
Query Syntax: Not Applicable
Default: Not Applicable

Last Modified:
01june2018 Added NF mode Opt. 356 (10.3)

**MMEMory:STORe:ENR <string>,<char>**

*(Write-Only)* Store a noise figure ENR (*.enr) file to the specified device.

- ENR files are saved and recalled from the "NoiseFigure\ENR" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

Note: Template *.enr files for several common models of noise sources may be found in the default ENR table directory of the FieldFox. In these files, the reflection coefficients have been set to the values specified for the respective model numbers, but the values for the ENR or ENR uncertainty will need to be entered at each frequency.

See also, MMEMory:LOAD:ENR.
See ALL MMEM commands.

Relevant Modes: **NF**

Parameters

- `<string>` For *.enr files, provide filename of the *.enr file WITHOUT an extension.
- `<char>` Memory device where the ENR file is stored. Choose from the following:
  - **INTernal** - internal storage device
  - **USB** - first USB device
  - **SD** - SD storage card

Examples: `MMEM:STOR:ENR "MySNR", INT`

Query Syntax: Not Applicable
Default: Not Applicable

Last Modified:
01june2018 Added NF mode Opt. 356 (10.3)

**MMEMory:STORe:FDATa <filename>**

*(Write-only)* Saves the selected formatted trace to a CSV or KML file to a default folder. (i.e., this is a screen capture from the currently displayed trace.)

See also, MMEM:STOR:MAT, MMEM:STOR:SDF, and MMEM:STOR:TXT.

Relevant Modes: **5GTF, CAT, IQA, LTE FDD, NA, NF, RTSA, SA, Pulse Measurements**

Parameters
```
<filename> Filename and extension into which data will be loaded.

Examples

```
Filename and extension into which data will be loaded.

Examples

```
MMEM:STOR:FDAT "MyFile.csv"

Query Syntax

Not Applicable

Default

Not Applicable

Last Modified:

<table>
<thead>
<tr>
<th>Date</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>03dec2018</td>
<td>Added LTE FDD &amp; 5GTF (A.1.0)</td>
</tr>
<tr>
<td>01june2018</td>
<td>Added NF mode Opt. 356 (10.3)</td>
</tr>
<tr>
<td>21nov2017</td>
<td>Added IQA mode (A.10.15)</td>
</tr>
<tr>
<td>20-sep-2016</td>
<td>Added RTSA mode (A.09.50)</td>
</tr>
<tr>
<td>31-Oct-2013</td>
<td>Added Pulse</td>
</tr>
<tr>
<td>23-Apr-2012</td>
<td>Major modifications</td>
</tr>
</tbody>
</table>

```
:MMEM:STOR:IMAGE <string>

(Write-only) Saves the current FieldFox screen to a *.png (picture) file. This file can NOT be read by the FieldFox.

Relevant Modes

ALL

Parameters

<string> Filename and extension of the *.png file.

Examples

```
MMEM:STOR:IMAGE "MyPic.png"

See an example using this command.

Query Syntax

Not Applicable

Default

Not Applicable

```

```
:MMEM:STOR:IMAGE:NOKeys <string>

(Write-only) Command to store full-screen images (without the softkeys)

Relevant Modes

ALL

Parameters

<string> Filename and extension of the *.png file.

Examples

```
MMEM:STOR:IMAGE:NOKeys "MyPic.png"

See an example using this command.

Query Syntax

Not Applicable

Default

Not Applicable

```

**MMEMory:STORe:IQCcapture:DEVice**

* (Read-Write) Set and query the type of device storage type for IQA data capture (i.e., INT, USB, or SD).

Use **MMEM:STOR:IQC:FTYP** to set the data capture file type.

Use **MMEM:STOR:IQC:FCO** to manually set the IQA file capture count.

**Relevant Modes** IQA

**Parameters**

<"device"> Data capture device storage-type.

- **INT** - internal FieldFox memory
- **SD** - SD card
- **USB** - USB storage device

**Examples** MMEM:STOR:IQC:DEV SD

**Query Syntax** MMEM:STOR:IQC:DEV?

**Return Type** character

**Default** INT

*Last Modified: 22-Oct-2017 Added new IQA mode content (10.1x).*

---

**MMEMory:STORe:IQCapture:FCOunt**

* (Read-Write) Set and query number of data capture files.

Use **MMEM:STOR:IQC:FCO:MULT** to enable or to disable the continuous mode file storage to an external SD card or USB storage device.

**Relevant Modes** IQA

**Parameters**

<num> Capture file count.

- **Preset:** 1
- **Minimum:** 1
- **Maximum:** 1000000000 (1e9)

**Examples** MMEM:STOR:IQC:FCO 1e9

**Query Syntax** MMEM:STOR:IQC:FCO?

**Return Type** Numeric
**Default**  1

---

**Last Modified:**

22-Oct-2017  Added new IQA mode content (10.1x).

---

**MMEMory:STORe:IQCapture:FCOunt:MULTiple <bool>**

*(Read-Write)* Set and query to enable or disable multiple (continuous) IQA file captures. Use **MMEMory:STORe:IQCapture:FCOunt** to manually set the IQA file capture count.

**Relevant Modes**  IQA

**Parameters**

< boolean >  Capture file count.

ON (1): Enables multiple (continuous) IQA file captures
OFF (0): Disables multiple (continuous) IQA file captures

**Examples**

```
MMEM:STOR:IQC:FCO:MULT ON
MMEM:STOR:IQC:FCO:MULT 0
```

**Query Syntax**  MMEM:STOR:IQC:FCO:MULT?

**Return Type**  boolean

**Default**  OFF (0)

---

**Last Modified:**

22-Oct-2017  Added new IQA mode content (10.1x).

---

**MMEMory:STORe:IQCapture:FNAMe <"string">**

*(Read-Write)* Set and query the filename for IQA data capture. Use **MMEM:STOR:IQC:FTYP** to set the data capture file type.

**Relevant Modes**  IQA

**Parameters**

<"string">  Data capture filename.

**Examples**

```
MMEM:STOR:IQC:FNAM "filename"
```

**Query Syntax**  MMEM:STOR:IQC:FNAM?

**Return Type**  character

**Default**  IQDataFile
MMEMory:STORe:IQCapture:FTYPe <char>

(Read-Write) Set and query the type of file type for IQA data capture (i.e., CSV, TXT, SDF, or MAT).
Use MMEM:STOR:IQC:FCO to manually set the IQA file capture count.
Use MMEM:STOR:IQC:DEV to set and query the type of device storage.

Relevant Modes  IQA

Parameters

<dataType>  Data capture file-type.
  CSV - csv file
  TXT - txt file
  SDF - sdf file
  MAT - matlab file

Examples

MMEM:STOR:IQC:FTYP MAT

Query Syntax

MMEM:STOR:IQC:FTYP:MULT?

Return Type  character

Default  CSV

Last Modified:
22-Oct-2017  Added new IQA mode content (10.1x).

MMEMory:STORe:IQCapture:STARt

(Write Only) Starts the IQA data capture.
Use SENS:MEAS:CAPT:TIME to set the data capture time or use MEAS:CAPT:LENGTH to set capture samples for the IQA data.

Relevant Modes  IQA

Parameters

n/a

Examples

MMEM:STOR:IQC:STAR

Query Syntax  n/a
Return Type  n/a
Default  n/a

Last Modified:
22-Oct-2017  Added new IQA mode content (10.1x).

**MMEMory:STORe:IQCapture:STOP**

*(Write Only)* Stops the IQA data capture.

Use `SENS:MEAS:CAPT:TIME` to set the data capture time or use `MEAS:CAPT:LENGTH` to set capture samples for the IQA data.

**Relevant Modes**  IQA

**Parameters**

n/a

**Examples**  

n/a

**Query Syntax**  

`MMEM:STOR:IQC:STOP`

**Return Type**  n/a

**Default**  n/a

Last Modified:
22-Oct-2017  Added new IQA mode content (10.1x).

**:MMEMory:STORe:LOG:CSV**

*(Read-Write)* Saves a CSV file to the Channel Scanner folder on customer-selected device: Int, USB, SD card.

**Relevant Modes**  ChScn

**Parameters**

`<filename>`

**Examples**  

`:MMEMory:STORe:LOG:CSV 'samplefile'`

**Query Syntax**  

`:MMEMory:STORe:LOG:CSV?`

**Return Type**  character

**Default**  

**:MMEMory:STORe:LOG:KML**

*(Read-Write)* Saves a KML file to the Channel Scanner folder on customer-selected device: Int, USB, SD card.

**Relevant Modes**  
ChScn

**Parameters**

<filename>

**Examples**  
:MMEMory:STORe:LOG:KML 'samplefile'

**Query Syntax**  
:MMEMory:STORe:LOG:KML?

**Return Type**  
character

**Default**

Last Modified:

---

**MMEMory:STORe:MAT <string>**

*(Write-only)* Save I/Q data as Matlab (mat) file. Only save file in one time. I/Q screen data cannot be recalled.

See also, MMEM:STOR:FDAT, MMEM:STOR:SDF, and MMEM:STOR:TXT.

**Relevant Modes**  
IQA

**Parameters**

<string>  
Filename of the mat file.

**Examples**  
MMEM:STOR:MAT "MyMatFile"

**Query Syntax**  
N/A

**Default**  
N/A

Last Modified:

19-Dec-2018  
New command

---

**MMEMory:STORe:PAMPliifier <string>,<char>**

*(Write-Only)* Store an uncertainty preamplifier (*.amp) file to the specified device.
• Preamplifier files (*.amp) are saved and recalled from the "NoiseFigure\AMP" folder. If the folder does not already exist on a USB or SD card, it is created automatically before storing the file.

Note: Template *.amp files for several common models of preamplifiers may be found in the default AMP table directory of the FieldFox. In these files, the reflection coefficients have been set to the values specified for the respective model numbers, but the values for the AMP or AMP uncertainty will need to be entered at each frequency.

See also, MMEMory:LOAD:PAMPlifier and MMEMory:IMPort:PAMPlifier.

See ALL MMEM commands.

Relevant Modes NF

Parameters
<string> For *.amp files, provide filename of the *.amp file WITHOUT an extension.
<char> Memory device where the preamplifier file is stored. Choose from the following:
• INTernal - internal storage device
• USB - first USB device
• SD - SD storage card

Examples MMEM:STOR:PAMP "MyAMP", INT

Query Syntax Not Applicable

Default Not Applicable

Last Modified:
01june2018 Added NF mode Opt. 356 (10.3)

MMEMory:STORe:SDF <string>

(Write-only) Save I/Q data as sdf file. Only save file in one time. I/Q screen data cannot be recalled.

See also, MMEM:STOR:FDAT, MMEM:STOR:TXT, and MMEM:STOR:MAT.

Relevant Modes IQA

Parameters
<string> Filename of the sdf file.

Examples MMEM:STOR:SDF "MySdfFile"

Query Syntax N/A

Default N/A

Last Modified:
19-Dec-2018 New command
MMEMory:STORe:SNP[:DATA] <filename>
(Write-only) Saves the selected trace to an SNP file.
See ALL MMEM commands.

Relevant Modes CAT, NA

Parameters

<string> Filename and extension to which data will be saved.
When saving 1-port data, use *.s1p
When saving 2-port data, use *.s2p

Examples

With S11 trace active:
MMEM:STOR:SNP "MyFile.s1p"

With S21 trace active:
MMEM:STOR:SNP "MyFile.s2p"

Query Syntax Not Applicable

Default Not Applicable

Last Modified:
23-Apr-2012 Major modifications

MMEMory:STORe:STATe <string>
(Write-only) Saves the current settings to an instrument state file.

Relevant Modes ALL

Parameters

<string> Filename and extension of the state file.

Examples

MMEM:STOR:STATe "MyStateFile.sta"

Query Syntax Not Applicable

Default Not Applicable

MMEMory:STORe:TXT<string>,<char>
(Write-only) Save I/Q data as text (TXT) file. Only save file in one time. I/Q screen data cannot be recalled.
See also, MMEM:STOR:FDAT, MMEM:STOR:SDF, and MMEM:STOR:MAT.

Relevant Modes IQA

Parameters

<string> Filename of the txt file.
Examples
MMEM:STOR:TXT "MyTxtFile"

Query Syntax
N/A

Default
N/A

Last Modified:
19-Dec-2018  New command

:RECPlayback:ACTion:PAUSE

(Write-Only) Recording or playback is temporarily halted. When the command is resent, recording or playback is resumed. A session should first be opened and recording or playing. Otherwise, this command is ignored.

Relevant Modes  LTE FDD, 5GTF, SA, RTSA
Parameters  None
Examples  RECP:ACT:PAUS
Query Syntax  Not Applicable
Default  Not Applicable

Last Modified:
27-nov-2018  Added LTE FDD and 5GTF mode (11.0)
10-june-2016  Added RTSA mode (9.50)

:RECPlayback:ACTion:PLAY

(Write-Only) Plays the current Record/Playback session. (Opt 236). A session with records should first be opened and stopped. Otherwise, this command is ignored.

Relevant Modes  LTE FDD, 5GTF, SA, RTSA
Parameters  None
Examples  RECP:ACT:PLAY
Query Syntax  Not Applicable
Default  Not Applicable

Last Modified:
27-nov-2018  Added LTE FDD and 5GTF mode (11.0)
10-june-2016  Added RTSA mode (9.50)
:RECPlayback:ACTion:POSition <num>

(Read-Write) Set and return a component carrier position. Use this command to specify a component carrier number when playback is paused.

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

- `<num>` Enter a value between 1 and the total recorded result:
  - 10000 maximum

**Examples**

RECP:ACT:POS 2

**Query Syntax**

:RECPlayback:ACTion:POSition?

**Default**  1

---

Last Modified:

19-Dec-2018  New command

:RECPlayback:ACTion:POSition:AUTO <string/num>

(Read-Write) Set and return a recorded result position. Use this command to move to a specific recorded result number when playback is paused.

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

- `<num>`
  - 0 (OFF) - disables AUTO and the component carrier is not random
  - 1 (ON) - enables AUTO and a random component carrier number is selected when playing is paused.

**Examples**

RECP:ACT:POS:AUTO 0
RECP:ACT:POS:AUTO ON

**Query Syntax**

:RECPlayback:ACTion:POSition:AUTO?

**Default**  1 (ON)

---

Last Modified:

19-Dec-2018  New command

:RECPlayback:ACTion:RECord

(Write-Only) Begins or resumes recording. (Opt 236).

**Relevant Modes**  LTE FDD, 5GTF, SA, RTSA
Parameters: None
Examples: RECP:ACT:REC
Query Syntax: Not Applicable
Default: Not Applicable

Last Modified:
27-nov-2018 Added LTE FDD and 5GTF mode (11.0)
10-june-2016 Added RTSA mode (9.50)

:RECPPlayback:ACTion:SAVE

(Write-Only) Saves the current data as a *.kml (default) or *.csv file.

Relevant Modes: LTE FDD, 5GTF
Parameters: None
Examples: RECP:ACT:SAVE
Query Syntax: Not Applicable
Default: Not Applicable

Last Modified:
19-Dec-2018 New command

:RECPPlayback:ACTion:SPOSition <num>

(Read-Write) Set and return a state position. Use this command to move to a specific state number when playback is paused.

Relevant Modes: SA, RTSA
Parameters: <num>
Enter a value between 1 and the total number of state tags.
Examples: RECP:ACT:SPOS 2
Query Syntax: :RECPPlayback:ACTion:SPOSition?
Default: Not Applicable

Last Modified:
10-june-2016 Added RTSA mode (9.50)
:RECPlayback:ACT:ION:STOP

(Write-Only) Stops recording or playback or a Record/Playback session. (Opt 236). A session must first be opened and recording, playing, or paused. Otherwise, this command is ignored.

**Relevant Modes**  SA, RTSA

**Parameters**  None

**Examples**  RECP:ACT:STOP

**Query Syntax**  Not Applicable

**Default**  Not Applicable

**Last Modified:**

27nov2018  Added LTE FDD and 5GTF mode (11.0)

10-june-2016  Added RTSA mode (9.50)

:RECPlayback:ACT:ION:TPOS:ITION <num>

(Read-Write) Set and return the current trace position. Use this command to move to a specific trace recording number when playback is paused.

**Relevant Modes**  SA, RTSA

**Parameters**

<num>  Enter a value between 1 and the total number of recorded traces.

**Examples**  RECP:SESS:TPOS 2

**Query Syntax**  :RECPlayback:ACT:ION:TPOS:ITION?

**Default**  Not Applicable

**Last Modified:**

10-june-2016  Added RTSA mode (9.50)


(Read-Write) Set and query the data storage device type (INTernal, USB, or SD).

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**
Available device types:

- **USB**  First (detected) USB flash drive is used to store data.
- **SD**  SD storage card
- **INT**  INTernal FieldFox storage

**Examples**

- RECPlayback:CONFig:FILE:DEVice USB
- RECPlayback:CONFig:FILE:DEVice SD
- RECPlayback:CONFig:FILE:DEVice INT

**Query Syntax**

- RECPlayback:CONFig:FILE:DEVice?

**Return Type**

- <string>

**Default**  INTernal

---

**:RECPlayback:CONFig:FILE:OWRite <string/boolean>**

*(Read-Write)* Set and query the status of the Record Playback file overwrite command.

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

- <string/boolean>  Available file types:

  - **ON (1)**  Enable file overwrite to overwrite the mode's default filename (e.g., for LTE FDD and 5GTF the filename is: OTARecorder).
  - **OFF (0)**  Disables the Playback Recorder's file overwrite command. Storing Playback Record data, requires manual naming of the data filename.

**Examples**

- RECPlayback:CONFig:FILE:OWRite 1
- RECPlayback:CONFig:FILE:OWRite OFF

**Query Syntax**  RECPlayback:CONFig:FILE:OWRite?

**Return Type**  <boolean>

**Default**  1

---

Last Modified:

19-Dec-2018  New command for OTA modes
:RECPlayback:CONFig:FILE:TYPE <string>

(Read-Write) Set and query the data type (KML (default) or CSV).

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

<string> Available file types:

- **KML**  Keyhole Markup Language
- **CSV**  Comma-Separated Values

**Examples**

RECPlayback:CONFig:FILE:TYPE KML
RECPlayback:CONFig:FILE:TYPE CSV

**Query Syntax**  
RECPlayback:CONFig:FILE:TYPE?

**Return Type**  <string>

**Default**  KML

Last Modified:  
19-Dec-2018  New command

:RECPlayback:CONFig:INTerval:DISTance <num>

(Read-Write) Set and query record distance interval. Only available when the measurement interval is set to DISTance. See **RECP:CONF:INT:TYPE**.

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

<num> Available choices:

- **Preset**  10
- **Range**  1 to 10000

**Examples**

RECPlayback:CONFig:INTerval:DISTance 5500
RECPlayback:CONFig:INTerval:DISTance 1

**Query Syntax**  
RECPlayback:CONFig:INTerval:DISTance?

**Return Type**  <num>

**Default**  10

Last Modified:  
19-Dec-2018  New command
:RECPlayback:CONFig:INTerval:STATE <string/boolean>

(Read-Write) Set and query the status of the interval requirement for saving records.

Relevant Modes  LTE FDD, 5GTF

Parameters

<string/boolean> Available file types:

ON (1) Enable records to be only saved when they meet the Interval requirement. See RECP:CONF:INT:TYPE.

OFF (0) Disables the interval requirement for saving records.

Examples

RECPlayback:CONFig:INTerval:STATE 1
RECPlayback:CONFig:INTerval:STATE OFF

Query Syntax  RECPlayback:CONFig:INTerval:STATE?

Return Type  <boolean>

Default  0

Last Modified:

19-Dec-2018 New command

:RECPlayback:CONFig:INTerval:TIME <num>

(Read-Write) Only available when the measurement interval is set to TIME. See RECP:CONF:INT:TYPE.

Relevant Modes  LTE FDD, 5GTF

Parameters

<num> Available choices:

Preset  20 second

Range  1 to 999 seconds

Examples

RECPlayback:CONFig:INTerval:TIME 50
RECPlayback:CONFig:INTerval:TIME 1

Query Syntax  RECPlayback:CONFig:INTerval:TIME?

Return Type  <num>

Default  20

Last Modified:

19-Dec-2018 New command
:RECPlayback:CONFig:INTerval:TYPE <string>

(Read-Write) Set and query the measurement interval for saving records type (DISTance or TIME (default)).

Relevant Modes  
LTE FDD, 5GTF

Parameters
<string>  Available file types:

TIME  Sets the measurement interval type for saving records to TIME.

DIST  Sets the measurement interval type for saving records to DISTance.

Examples
RECPlayback:CONFig:INTerval:TYPE TIME
RECPlayback:CONFig:INTerval:TYPE DIST

Query Syntax
RECPlayback:CONFig:INTerval:TYPE?

Return Type <string>

Default TIME

Last Modified:
19-Dec-2018  New command

:RECPlayback:CONFig:FMTRigger:DATA <data> - Superseded

Note: This command is replaced by RECPlayback:CONFig:FMTRigger:LLData which can be used with Relative masks.

(Read-Write) Set and query the data to complete the Frequency Mask Trigger (FMT) table, used for Record Playback. Measured power levels above an upper limit and below a lower limit will trigger recording.


Relevant Modes  
SA

Parameters
<data> Data for all FMT segments in the following format:
  
  - \( n \) = number of segments, followed by segment data.
  
  - each segment: \textbf{State}, \textbf{Type}, \textbf{BegStim}, \textbf{EndStim}, \textbf{BegResp}, \textbf{EndResp}
  
  - Where:
    
    \textbf{State} 0 for limit line disabled
    
    1 for limit line enabled.
    
    \textbf{Type} Type of limit segment. Choose from:
    
    0 - Upper limit
    
    1 - Lower limit
    
    \textbf{BegStim} Start of X-axis value (freq, power, time)
    
    \textbf{EndStim} End of X-axis value
    
    \textbf{BegResp} Y-axis value that corresponds with Start of X-axis value
    
    \textbf{EndResp} Y-axis value that corresponds with End of X-axis value
    
  Subsequent segments are appended to the data in the same manner.
  
  \textbf{Examples} The following writes three upper limit segments.
  
  'individual segments are colored for readability.
  
  \texttt{RECP:CONFig:FMTR:DATA 3,1,0,2e7,3e7,-}
  
  \texttt{30,0,1,0,3e7,5e7,0,0,1,0,5e7,6e7,0,-30}

  \textbf{Query Syntax} \texttt{RECP:CONFig:FMTR:DATA?}
  
  \textbf{Return Type} Block data
  
  \textbf{Default} 0 - Limit line data off

Last Modified:

16-Aug-2012 Superseded A.06.00

:\texttt{RECP:CONFig:FMTR:ENABle <bool>}

\textbf{(Read-Write)} Set and return the state of Frequency Mask Triggering. Use \texttt{RECP:CONFig:FMTR:DATA} to construct Frequency Mask Trigger limits.

\textbf{Relevant Modes} SA

\textbf{Parameters}

\texttt{<bool>} Frequency Mask Trigger state. Choose from:

\texttt{OFF} or 0 - Frequency Mask Trigger disabled.

\texttt{ON} or 1 - Frequency Mask Trigger enabled.

\textbf{Examples} \texttt{RECP:CONFig:FMTR:ENABle 1}
Query Syntax  :RECPlayback:CONFig:FMTRigger:ENABle?

Default  OFF

:RECPlayback:CONFig:FMTRigger:LLData <data>

Note: This command replaces :RECPlayback:CONFig:FMTRigger:DATA which can be used ONLY with a Fixed mask. This command can also be used with Relative masks.

(Read-Write) Set and query the data to complete the Frequency Mask Trigger (FMT) table, used for Record Playback. Measured power levels above an upper limit and below a lower limit will trigger recording.


Relevant Modes  SA

Parameters

<data>  Data for all FMT segments in the following format:
  •  \( \tilde{n} \) = number of segments, followed by segment data.
  •  Where:

  \( <nL> \)  number of FMT segments to follow

  State  0 - segment disabled
          1 - segment enabled.

  Fixed/Rel  0 - Relative
             1 - Fixed

  Upper/Lower  0 - Upper segment
                1 - Lower segment

  \( <nP> \)  Number of points to follow

  Freq value  X-axis value

  Amp value  Y-axis value

Subsequent points are appended to the data in the same manner.

Examples
"The following writes three upper segments.
individual segments are colored for readability."  
:RECPlayback:CONFig:FMTRigger:LLData 1,1,0,0,4,-30e6,-20,-20e6,-10,-10e6,0,10e6,0,20e6,-10,30e6,-20

Query Syntax  :RECPlayback:CONFig:FMTRigger:LLData?

Return Type  Block data

Default  0 - Limit line data off

Last Modified:
16-Aug-2012 New command

:RECPlayback:CONF:PTINterval <num>

(Read-Write) Set and return the Playback Time Interval - the delay that occurs between each trace as it is played back.

Relevant Modes SA, RTSA

Parameters

<num> Playback Time Interval (in seconds). Choose a delay value between 0 (play as fast as possible) and 100.

Examples RECP:CONF:PTIN 10

Query Syntax :RECPlayback:CONF:PTINterval?

Default 0

Last Modified: 10-june-2016 Added RTSA mode (9.50)

:RECPlayback:CONF:RSCLength <num>

(Read-Write) Set and return the Record Segment Counting Length - the number of traces to record, after which the recording will automatically pause. Use Pause/Resume or Record to capture another <num> traces, or Stop to end recording.

When set, a counter appears in the lower-left corner which counts UP to the specified number of recordings.

Relevant Modes SA, RTSA

Parameters

<num> Number of traces to record. Choose a value between 0 (no limit to the number of traces) and 100.

Examples RECP:CONF:RSCL 10

Query Syntax :RECPlayback:CONF:RSCLength?

Default OFF - NO limit to the number of traces to be recorded.

Last Modified: 10-june-2016 Added RTSA mode (9.50)

:RECPlayback:CONF:RSource <char>
(Read-Write) Set and return the Record Source - the location in the FieldFox data flow from where data is recorded.

**Relevant Modes**  SA, RTSA

**Parameters**

- `<num>` Record Source. Choose from:
  - **FATR** - First Active Trace. Means the first active trace that is not set to BLANK and or not set to VIEW. (RTSA mode only)
  - **RAWM** - Records raw measurement data (SA mode only)
  - **TRA1** - Trace 1
  - **TRA2** - Trace 2
  - **TRA3** - Trace 3
  - **TRA4** - Trace 4

**Examples**

```
RECP:CONF:RSO RAWM
```

**Query Syntax**

`:RECPlayback:CONFig:RSOurce?`

**Default**

- FATR (RTSA mode default)
- RAWM (SA mode default)

---

Last Modified:

10-june-2016  Added RTSA mode (9.50)

---

`:RECPlayback:CONFig:RTINerval <num>`

(Read-Write) Set and return the Record Time Interval - the delay between each recording.

**Relevant Modes**  SA, RTSA

**Parameters**

- `<numeric>` Record Time Interval (in seconds). Choose a delay value between 0 (record as fast as possible) and 10,000.

**Examples**

```
RECP:CONF:RTIN 10
```

**Query Syntax**

`:RECPlayback:CONFig:RTINterval?`

**Default**

0

---

Last Modified:

10-june-2016  Added RTSA mode (9.50)

---

`:RECPlayback:CONFig:RTLSeconds <num>`

(Read-Write) Set and return the Record Time limit seconds - the time duration for the recording, after which the recording will automatically pause. Use Pause/Resume to capture another `<num>` seconds, or
Stop to end recording.
When set, a timer appears in the lower-left corner which counts DOWN from the specified number of seconds.

**Relevant Modes**  SA, RTSA

**Parameters**

<numeric>  Record Time limit (in seconds). Choose a delay value between 0 (NO limit - MAX) and 1e6 seconds.

**Examples**  `RECP:CONF:PLTS 100`

**Query Syntax**  `:RECPPlayback:CONFig:RTLSeconds?`

**Default**  0 - NO limit to the number of seconds for a recording.

---

**Last Modified:**
10-june-2016  Added RTSA mode (9.50)

---

`:RECPPlayback:MACHine:TPOSition?`

(Read-Only) Can be issued during Playback to yield the same trace position information that is currently displayed on-screen .

**Relevant Modes**  SA, RTSA

**Parameters**  None

**Examples**  `RECP:MACH:/TPOS?`

**Default**  Not Applicable

---

**Last Modified:**
19-Dec-2018  New command

---

`:RECPPlayback:MACHine:TTIMe?`

(Read-Only) Can be issued during Playback to yield the same trace time information that is currently displayed on-screen .

**Relevant Modes**  SA, RTSA

**Parameters**  None

**Examples**  `RECP:MACH:TTIM?`

**Default**  Not Applicable
### :RECPlayback:SESSION:CARRecords

**(Write-Only)** Clears all recorded traces from the currently-open Record/Playback session. (Opt 236)

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA, RTSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td><code>&lt;string&gt;</code> Filename and extension of the state file.</td>
</tr>
<tr>
<td>Example</td>
<td><code>RECP : SESS : CAR</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

**Last Modified:**
19-Dec-2018      New command

**Last Modified:**
10-June-2016     Added RTSA mode (9.50)

### :RECPlayback:SESSION:CLOSESe

**(Write-Only)** Closes the currently-open Record/Playback session. (Opt 236). The session can be opened again for recording or playback using :RECPlayback:SESSION:OPEN?

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA, RTSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td><code>RECP : SESS : CLOS</code></td>
</tr>
<tr>
<td>Query Syntax</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

**Last Modified:**
10-June-2016     Added RTSA mode (9.50)

### :RECPlayback:SESSION:NEW

**(Write-Only)** Creates and opens a new Record/Playback session with an auto-generated name "AutoSession xx". (Opt 236).

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA, RTSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td><code>RECP : SESS : NEW</code></td>
</tr>
</tbody>
</table>

**Last Modified:**
10-June-2016     Added RTSA mode (9.50)
Query Syntax: :RECPlayback:SESSion:OPEN? <"string">

(Read-Only) Opens an existing Record/Playback session and returns 2 numbers separated by a comma.
1. The current quantity of traces in the session.
2. The current quantity of states in the session.
   • -1, -1 means nothing was opened. A session may already be open.
   • 0, 0 means the session was opened but currently empty.

Relevant Modes: SA, RTSA

Parameters: None

<"string"> Name of the session to be opened, enclosed in quotes. Not case sensitive.

Examples:

RECP:SESS:OPEN? "MyOldSession"

Default: Not Applicable

Last Modified:
10-june-2016 Added RTSA mode (9.50)

:RECPlayback:SESSion:SDEVice <char>

(Read-Write) Storage device to be used for record playback. (Opt 236)

Relevant Modes: SA, RTSA

Parameters:

<character> Storage device. Choose from:
             INTernal - Internal storage
             SD - SD card
             USB - USB device

Examples:

RECP:SESS:SDEV USB

Query Syntax: RECPlayback:SESSion:SDEVice?

Default: INTernal

Last Modified:
10-june-2016 Added RTSA mode (9.50)
:RECP:PLAYback:SESSION:TRLIMIT <num>

(Read-Write) Trace record limit for the current record playback session (Opt 236). An upper limit is set when the session opens. You can reduce this value to save storage space, but you cannot increase it beyond the upper default limit.

**Relevant Modes** SA, RTSA

**Parameters**

<numeric> Trace record limit. Choose a value from 1 to the default upper limit. Send a query before setting this value to learn the upper limit.

**Examples**

RECP:SESSION:TRLIMIT 100

**Query Syntax**

RECP:SESSION:TRLIMIT?

**Default** Not Applicable

Last Modified:

10-june-2016 Added RTSA mode (9.50)

[:SENSe]:ACP:POWER:LIMIT[:STATe] <bool>

(Read-Write) Set and query the limit testing On/Off state in an Adjacent Channel Power measurement. To establish limits, use:

[:SENSe]:ACP:POWER:OFFSET:LLIMIT (Lower limit)
[:SENSe]:ACP:POWER:OFFSET:ULIMIT (Upper limit)

**Relevant Modes** SA

**Parameters**

<bool> ACP limit testing ON | OFF state. Choose from:

0 or OFF - Limit testing OFF
1 or ON - Limit testing ON

**Examples**

ACP:LIMIT ON

**Query Syntax**

[:SENSe]:ACP:LIMIT[:STATe]?

**Return Type** Boolean

**Default** OFF

[:SENSe]:ACP:POWER:MREFerence <num>

(Read-Write) Set and query the reference value for the measured offset power in an Adjacent Channel Power measurement. See also: [:SENSe]:ACP:POWER:MREFerence:AUTO

**Relevant Modes** SA
Parameters

<num>  Reference value in dB or dBC.

Examples  ACP:MREF -1.5

Query Syntax  [:SENSe]:ACPower:MREF?

Return Type  Numeric

Default  0

Last Modified:

16-Aug-2012  New command

[:SENSe]:ACPower:MREFerence:AUTO <bool>

(Read-Write) Set and query the method of computing the measured carrier power from the measured offset power in an Adjacent Channel Power measurement.

Relevant Modes  SA

Parameters

<bool>  Choose from:

1 or ON (Automatic) - dB or dBC value is computed by subtracting the measured carrier power from the measured offset power.

0 or OFF (Manual) - dB or dBC value is computed by subtracting the entered Ref Value from the measured offset power. Use [:SENSe]:ACPower:MREFerence to enter a value.

Examples  ACP:MREF:AUTO 1

Query Syntax  [:SENSe]:ACPower:MREFerence:AUTO?

Return Type  Boolean

Default  ON

Last Modified:

16-Aug-2012  New command

[:SENSe]:ACPower:MTYPe <char>

(Read-Write) Set and query the ACP measurement type. This setting determines how the measured carrier and offset power levels are presented.

Relevant Modes  SA

Parameters
### Measurement type. Choose from:

- **TPWR** - Total Power Ref
- **PSDR** - Power Spectral Density

### Examples

- ACP:MREF -1.5

### Query Syntax

```
[:SENSe]:ACPower:MTYPe?
```

### Return Type

Character

### Default

TPWR

---

**Last Modified:**

16-Aug-2012 New command

---

#### [:SENSe]:ACPower:OFFSet<n>:BWIDth <num>

*(Read-Write)* Set and query the integration bandwidths for the 3 offset channels in an Adjacent Channel Power measurement.

**Relevant Modes**

SA

**Parameters**

- `<n>` Offset number to be set. Choose from 1, 2, or 3.
- `<num>` Offset Integration Bandwidth in Hz. Choose a number between the Minimum Integration Bandwidth (100 Hz) and Maximum Integration Bandwidth (100 MHz)

**Examples**

- ACP:OFFS1:BWID 2e6
- Sets integration bandwidth of offset 1 to 2 MHz

**Query Syntax**

```
[:SENSe]:ACPower:OFFSet:BWIDth?
```

**Return Type**

Numeric

**Default**

2 MHz

---

#### [:SENSe]:ACPower:OFFSet<n>:FREQuency <num>

*(Read-Write)* Set and query the Offset frequencies for all 3 offset channels in an Adjacent Channel Power measurement.

**Relevant Modes**

SA

**Parameters**

- `<n>` Offset number to be set. Choose from 1, 2, 3.
- `<num>` Offset Channel Frequency in Hz. For each offset, choose a number between the Minimum Offset Frequency (0 Hz) and Maximum Offset Frequency (500 MHz).
Examples

ACP:OFFS1:FREQ 2e6
Sets offset 1 to 2 MHz

Query Syntax

[:SENSe]:ACPower:OFFSet:FREQuency?

Return Type

Numeric

Default

3 MHz, 0 Hz, 0 Hz

[:SENSe]:ACPower:OFFSet<n>:LLIMit <num>

(Read-Write) Set and query the limit for the 3 Lower offset channels in an ACP measurement. Power measurements that exceed this limit will FAIL. To turn limit testing ON and OFF, use [:SENSe]:ACPower:LIMit[:STATe].

Relevant Modes

SA

Parameters

<n> Offset number. Choose from 1, 2, or 3
<num> Limit value for Lower offsets in dBc. For each offset, choose a value between 500 and -1000.

Examples

ACP:OFFS1:LLIM -10
Sets the first lower offset limit value to -10 dBc.

Query Syntax

[:SENSe]:ACPower:OFFSet<n>:LLIMit?

Return Type

Numeric

Default

0 dBc for all offsets

Last Modified:

17-Aug-2012 New command- replaces CALC command

[:SENSe]:ACPower:OFFSet<n>:STATe <bool>

(Read-Write) Set and query the On/Off state for the 3 offset channels in an Adjacent Channel Power measurement.

Relevant Modes

SA

Parameters

<n> Offset number. Choose from 1, 2, 3.
<bool> ACP Limit Test ON | OFF state. For each offset, choose from:
0 or OFF - Offset OFF
1 or ON - Offset ON

Examples

ACP:OFFS1:STAT ON
Query Syntax: [:SENSe]:ACPower:OFFSet:STATe?

Return Type: Boolean
Default: ON, OFF, OFF

[:SENSe]:ACPower:OFFSet<n>:ULIMit <num>

(Read-Write) Set and query the limit for the 3 Upper offset channels in an ACP measurement. Power measurements that exceed this limit will FAIL. To turn limit testing ON and OFF, use [:SENSe]:ACPower:LIMit[:STATe].

Relevant Modes: SA

Parameters:

<n> Offset number. Choose from 1, 2, or 3
<num> Limit value for Upper offsets in dBc. For each offset, choose a value between 500 and -1000.

Examples:
ACP:OFFS1:ULIM -10
Sets the first upper offset limit value to -10 dBc.

Query Syntax: [:SENSe]:ACPower:OFFSet<n>:ULIMit?

Return Type: Numeric
Default: 0 dBc for all offsets

Last Modified: 17-Aug-2012  New command- replaces CALC command

[:SENSe]:ACQuisition:TIME

(Read-Write) Sets the acquisition time between Density and Real-Time traces. (Spectrogram is not supported.)

Relevant Modes: RTSA

Parameters:

<num> Range: Span dependent

Examples:
ACQ:TIME .05
ACQ:TIME 5.00E-1

Query Syntax: [:SENSe]:ACQuisition:TIME?

Return Type: numeric
Default: 2.00E-02
**[SENSe]:ACQuisition:TIME:AUTO <bool>**

*(Read-Write)* Sets the acquisition time between Density and Real-Time traces to AUTO. (Spectrogram is not supported.)

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>RTSA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>&lt;Boolean&gt;</td>
<td></td>
</tr>
<tr>
<td>Automatically sets the range: 20 ms (2.00E-02) to 541.2 ms (5.412E-1)</td>
<td></td>
</tr>
<tr>
<td>ON (1) - The FieldFox automatically sets the acquisition time.</td>
<td></td>
</tr>
<tr>
<td>OFF (0) - Auto acquisition is disabled.</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

```
ACQ:TIME:AUTO
```

**Query Syntax**

```
[:SENSe]:ACQuisition:TIME:AUTO?
```

**Return Type** numeric

**Default** 0

---

Last Modified:

19-sep-2016 Added new RTSA command (A.09.50).

---

**[:SENSe]:ADEMod:METRics:AMTY**

*(Read-Write)* Sets the AM window Y axis top percentage value (10-100%).

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>&lt;value&gt;</td>
<td>AM Y axis percentage</td>
</tr>
</tbody>
</table>

**Examples**

```
:ADEM:METR:AMTY 10
```

**Query Syntax**

```
:ADEM:METR:AMTY?
```

**Return Type** Value

**Default** 100

---

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)
[:SENSe]:ADEMod:METRics:DTYPe

(Read-Write) Select the type of metrics AM, FW wideband, or FM narrow band.

**Relevant Modes** SA

**Parameters**

- AM - Selects AM metrics
- FMN - Selects FM narrow band metrics
- FMW - Selects FM wide band metrics

**Examples**

:ADEM:METR:DTYP FMN

**Query Syntax**

:ADEM:METR:DTYP?

**Return Type** String

**Default**

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ADEMod:METRics:FMTY

(Read-Write) Set the FM window Y axis top percentage value (10-100%).

**Relevant Modes** SA

**Parameters**

- `<num>` FM Y axis percentage

**Examples**

:ADEM:METR:FMTY 25

**Query Syntax**

:ADEM:METR:FMTY?

**Return Type** Numeric

**Default** 100

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)
[:SENSe]:ADEMod:METRics:LON
(Read-Write) Enable or disable the audio while AM/FM metrics are on.

**Relevant Modes**  
SA

**Parameters**  
ON - Enable audio while metrics are on  
OFF - Disable audio while metrics are on

**Examples**  
:ADEM:METR ON

**Query Syntax**  
:ADEM:METR?

**Return Type**  
String

**Default**  
OFF

Last Modified:
16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ADEMod:METRics:LTIme
(Read-Write) Sets the listen time.

**Relevant Modes**  
SA

**Parameters**  
<num> Listen time value (100 ms to 100 seconds)

**Examples**  
:ADEM:METR:LTIM 100ms

**Query Syntax**  
:ADEM:METR:LTIM?

**Return Type**  
Numeric

**Default**  
100s

Last Modified:
16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ADEMod:METRics:MMEEnable
(Read-Write) Enable or disable of AM Mod Depth or FM Freq Deviation Peak+ and Peak- in the
demodulation window.

**Relevant Modes** SA

**Parameters**

- **ON** - Enables Peak+ and Peak-
- **OFF** - Disables Peak+ and Peak-

<string>

**Examples** :ADEM:METR:MMEN ON

**Query Syntax** :ADEM:METR:MMEN?

**Return Type** String

**Default** OFF

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ADEMod:METRics:STIme

(Read-Write) Sets the time span of the modulation window.

**Relevant Modes** SA

**Parameters**

<num> Modulation time value (50 us to 50 ms)

**Examples** :ADEM:METR:STIM .025

**Query Syntax** :ADEM:METR:LTIM?

**Return Type** Numeric

**Default** 2.000 ms

Last Modified:

16-Dec-2015 Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ADEMod:METRics:TFReq

(Read-Write) Sets the tune frequency (i.e., center frequency).

**Relevant Modes** SA
Parameters

<num>  Tune (Center) frequency values vary with the FieldFox.

Examples

:ADEM:METR:TFR  6GHZ

Query Syntax

:ADIM:METR:TFR?

Return Type

Numeric

Default

Varies with model

Last Modified:

16-Dec-2015  Added AM/FM Metrics (8.15 & 9.15)

[:SENSe]:ALIGnment:ALL:NOW

(Write-only)  An "InstAlign" alignment now is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered, and alignments are simultaneously performed on all of the individual alignments (i.e., InstAlign Amplitude Alignment, RF Burst, and Channel Equalization alignments). Refer to ALIGn:ALL.

Relevant Modes  IQA, NF, SA, CPM, RTSA

All models with InstAlign (All EXCEPT N9912A)

Parameters  None

Examples  ALIG:ALL:NOW

Query Syntax  Not Applicable

Default  Not Applicable

Last Modified:

01June2018  Added NF mode Opt. 356 (10.3)

19-Dec-2018  New command & new IQA mode (A.10.15)

[:SENSe]:ALIGnment:ALL[:STATe] <char>

(Read-Write)  Set and query the align all (InstAlign all) state. When the individual alignments are coupled (Align All state is set to Auto), the state of all of them can be changed simultaneously using the Align All (i.e., InstAlign Amplitude Alignment, RF Burst, and Channel Equalization alignments). Refer to ALIGn:ALL:NOW.

Relevant Modes  ChScn, IQA, LTE FDD, SA, CPM, RTSA

All models with InstAlign capability (All EXCEPT N9912A).
## Parameters

**<character>** InstAlign state. Choose from:

- **AUTO** - The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.
- **HOLD** - The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.
- **OFF** - The alignment process is NOT performed. Only factory correction values are used.

### Examples

```
ALIG:AMPL HOLD
```

### Query Syntax

```
[:SENSe]:ALIGnment:AMPLitude:NOW
```

### Return Type

Character

### Default

AUTO

---

Last Modified:

19-Dec-2018 New command

---

### [:SENSe]:ALIGnment:AMPLitude:NOW

(Write-only) An "InstAlign" alignment is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

#### Relevant Modes

- ChScn, IQA, LTE FDD, SA, CPM, RTSA
- All models with InstAlign capability (All EXCEPT N9912A)

#### Parameters

None

#### Examples

```
ALIG:AMPL:NOW
```

#### Query Syntax

Not Applicable

#### Default

Not Applicable

---

Last Modified:

19-Dec-2018 New command & new IQA mode (A.10.15)

---

### [:SENSe]:ALIGnment:AMPLitude[:STATe] <char>

(Read-Write) Set and query the amplitude alignment (InstAlign) state.

#### Relevant Modes

- ChScn, IQA, LTE FDD, SA, CPM, RTSA
- All models with InstAlign capability (All EXCEPT N9912A).
Parameters

<character> InstAlign state. Choose from:

AUTO - The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.

HOLD - The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.

OFF - The alignment process is NOT performed. Only factory correction values are used.

Examples ALIG:AMPL HOLD

Query Syntax [:SENSe]:ALIGNment:AMPLitude[:STATe]?

Return Type Character

Default AUTO

Last Modified:
19-Dec-2018 New command & new IQA mode (A.10.15)

[:SENSe]:ALIGNment:BURSt:NOW

(Write-only) A RF Burst alignment is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

Relevant Modes IQA, SA & RTSA Modes

All models with RF Burst triggering (All EXCEPT N9912A)

Parameters None

Examples ALIG:BURS:NOW

Query Syntax Not Applicable

Default Not Applicable

Last Modified:
19-Dec-2018 New command and new IQA mode (A.10.15)

[:SENSe]:ALIGNment:BURSt[:STATe]

(Read-Write) Set and query the RTSA and SA modes burst alignment (InstAlign) state.

Relevant Modes IQA, SA, CPM, RTSA

All models with InstAlign capability (All EXCEPT N9912A).
**Parameters**

<char> InstAlign state. Choose from:

- **AUTO** - The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.

- **HOLD** - The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.

- **OFF** - The alignment process is NOT performed. Only factory correction values are used.

**Examples** ALIG:BURS HOLD

**Query Syntax** [:SENSe]:ALIGnment:BURSt[:STATe]?

**Return Type** Character

**Default** AUTO

---

**Last Modified:**

19-Dec-2018 New command & new IQA mode (A.10.15)

---

### [:SENSe]:ALIgnment:CHEQ:NOW

**(Write-only)** A channel equalization is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

**Relevant Modes** IQA, RTSA

**Parameters** None

**Examples** ALIG:CHEQ:NOW

**Query Syntax** Not Applicable

**Default** Not Applicable

---

**Last Modified:**

19-Dec-2018 New command & new IQA mode (A.10.15)

---

### [:SENSe]:ALIgnment:CHEQ[:STATe]

**(Read-Write)** Set and query the channel equalization (InstAlign) state.

**Relevant Modes** IQA, RTSA

**Parameters**
InstAlign state. Choose from:

- **AUTO** - The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.

- **HOLD** - The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.

- **OFF** - The alignment process is NOT performed. Only factory correction values are used.

**Examples**

```
ALIG:CHEQ HOLD
```

**Query Syntax**

```
[:SENSe]:ALIGnment:CHEQ[:STATe]?
```

**Return Type**

Character

**Default**

AUTO

---

Last Modified:

- 19-Dec-2018 New command & new IQA mode (A.10.15)

---

This command has been replaced by [:SENSe]AMPLitude:ALIGnment:NOW. Learn about superseded commands.

(Write-only) An "InstAlign" alignment is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

**Relevant Modes**

ChScn, IQA, LTE FDD, SA, CPM, RTSA

All models with InstAlign (All EXCEPT N9912A)

**Parameters**

None

**Examples**

```
AMPL:ALIG:NOW
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

Last Modified:

- 22-oct-2016 Added IQA mode and superseded note (A.010.15)
- 20-sep-2016 Added RTSA mode (A.09.50)
- 1-Apr-2014 Added CPM
- 17-Jul-2012 New command (6.00)
### [:SENSe]:AMPLitude:ALIGnment[:STATe] <char>

This command has been replaced by [:SENSe]:ALIGnment:AMPLitude[:STATe]. Learn about superseded commands.

**Read-Write** Set and query the alignment amplitude (InstAlign) state.

**Relevant Modes** IQA, SA, CPM, RTSA

All models with InstAlign capability (All EXCEPT N9912A).

**Parameters**

- `<character>`: InstAlign state. Choose from:
  - **AUTO**: The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.
  - **HOLD**: The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.
  - **OFF**: The alignment process is NOT performed. Only factory correction values are used.

**Examples**

- `:AMPL:ALIG HOLD`

**Query Syntax**

`:SENSe:AMPLi
tude:ALIGnment:STATe?`

**Return Type** Character

**Default** AUTO

---

Last Modified:

- 22-oct-2016 Added IQA mode and superseded note (A.010.15)
- 20-sep-2016 Added RTSA mode (A.09.50)
- 1-Apr-2014 Added CPM
- 28-Mar-2013 Changed time from 30 to 300 secs
- 17-Jul-2012 New command (6.00)

### [:SENSe]:AMPLitude:CORRections:ANTenna:DEFault

**Write-only** Clears the Receiver Antenna Correction table. This is the same as selecting the **New** softkey and then confirming “Yes” from the Edit/Save/Recall Antennas menu.

If using only one antenna correction, use this command.

For source antenna, use: [:SENSe]:AMPLitude:CORRections:SANTenna:DEFault

**Relevant Modes** ChScn, SA

**Parameters** None

**Examples** `AMPL:CORR:ANT:DEF`

**Query Syntax** Not Applicable
Default  Not Applicable

Last Modified:

01 June 2018  Added Channel Scanner (ChScn) - (10.3)
26-Jan-2015  Modified for Src/Rec
17-Jul-2012  New command (6.00)

[:SENSe]:AMPLitude:CORRections:ANTenna[:STATe] <bool>

(Read-Write) Set and query the Receiver Antenna Corrections ON/OFF state.
If using only one antenna correction, use this command.
For source antenna, use: [:SENSe]:AMPLitude:CORRections:SANTenna[:STATe]

Relevant Modes  ChScn, SA
Parameters

  <bool>  Antenna correction state. Choose from:
  OFF (or 0) - Antenna correction OFF.
  ON (or 1) - Antenna correction ON.

Examples  AMPL:CORR:ANT 1

Query Syntax  [:SENSe]:AMPLitude:CORRections:ANTenna[:STATe]?

Return Type  Boolean

Default  OFF

Last Modified:

01 June 2018  Added Channel Scanner (ChScn) - (10.3)
26-Jan-2015  Modified for Src/Rec
17-Jul-2012  New command (6.00)

[:SENSe]:AMPLitude:CORRections:CABLE:DEFault

(Write-only) Clears the Receiver Cable Correction table. This is the same as selecting the New softkey
and then confirming “Yes” from the Edit/Save/Recall Antennas menu.
If using only one cable correction, use this command. For source cable, use
[:SENSe]:AMPLitude:CORRections:SCABLE:DEFault

Relevant Modes  ChScn, SA
Parameters  None
Examples
AMPL:CORR:CABL:DEF

Query Syntax
Not Applicable

Default
Not Applicable

Last Modified:
01June2018 Added Channel Scanner (ChScn) - (10.3)
26-Jan-2015 Modified for Src/Rec
17-Jul-2012 New command (6.00)

[:SENSe]:AMPLitude:CORRections:CABLe[:STATe] <bool>

(Read-Write) Set and query the Receiver Cable Corrections ON/OFF state.
If using only one cable correction, use this command.
For source cable, use: [:SENSe]:AMPLitude:CORRections:SCABle[:STATe]

Relevant Modes  ChScn, SA

Parameters
<bool>  Cable correction state. Choose from:
OFF (or 0) - Cable correction OFF.
ON (or 1) - Cable correction ON.

Examples
AMPL:CORR:CABL 1

Query Syntax  [:SENSe]:AMPLitude:CORRections:CABLe[:STATe]?

Return Type  Boolean

Default  OFF

Last Modified:
01June2018 Added Channel Scanner (ChScn) - (10.3)
26-Jan-2015 Modified for source.
20-Oct-2010 New command (5.30)

[:SENSe]:AMPLitude:CORRections[:STATe] <bool>

(Read-Write) Set and query the Channel Scanner (ChScn) and SA mode corrections ON/OFF state.
This setting is labeled "Apply Corrections" on the FieldFox softkeys.
See all corrections commands
**Relevant Modes**  ChScn, SA

**Parameters**

<bool>  Correction state. Choose from:
OFF (or 0) - Correction OFF.
ON (or 1) - Correction ON.

**Examples**  AMPL:CORR 1

**Query Syntax**  [:SENSe]:AMPLitude:CORRections[:STATe]?

**Return Type**  Boolean

**Default**  OFF

---

Last Modified:

01june2018  Added Channel Scanner (ChScn) - (10.3)
20-Oct-2010  New command (5.30)

**[:SENSe]:AMPLitude:SCALe <char>**

*(Read-Write)* Set and query the scale type.

**Relevant Modes**  SA

**Parameters**

<char>  Scale type. Choose from:
LOG - units in dB
LIN - units in mV

**Examples**  AMPL:SCAL LIN

**Query Syntax**  [:SENSe]:AMPLitude:SCALe?

**Return Type**  Character

**Default**  LOG

---

**[:SENSe]:AMPLitude:UNIT <char>**

*(Read-Write)* Set and query the Display Units, regardless of the current Scale setting. The UNIT choice affects the following: Reference Level, Trigger Level, Limit Lines, Marker annotation, Channel Power, and Adjacent Channel Power.

This command affects the TRACe:DATA? return values.

Antenna correction units are available ONLY by loading an Antenna file that contains the desired units setting. Use MMEMory:LOAD:ANTenna.

**Relevant Modes**  ChScn, SA, CPM
Parameters

<char> Display Units with Antenna correction OFF, choose from:

- W - watts
- DBM - dBm

The following are NF and SA ONLY - not allowed for CPM

- DBMV - dB milliVolts
- DBUV - dB microvolts
- DBMA - dB milliAmps
- DBUA - dB microAmps
- V - volts
- A - amps

Examples AMPL:UNIT W

Query Syntax [:SENSe]:AMPLitude:UNIT?

Return Type Character

Default DBM

Last Modified:

01june2018 Added Channel Scanner (ChScn) - (10.3)
24-Mar-2014 Added CPM
20-Oct-2010 New command (5.30)

[:SENSe]:AVERage:CLEAR

Write-only) Resets sweep averaging to zero so that the next sweep performed will be back to AVG 1. This command does NOT trigger the sweep.

Relevant Modes CAT, NA, VVM

Parameters None

Examples AVER:CLEAR

Query Syntax Not Applicable

Default Not Applicable

Last Modified:

03Oct2017 Added command for VVM mode
23-Aug-2011 New command
[:SENSe]:AVERage:COUNt <num>

(Read-Write) Set and query the number of sweep averages.

**Relevant Modes**
All

**Parameters**

<number> Number of averages. Choose from: 1 to 10000.
Average count of 1 = No averaging.

**Examples**

AVER:COUN 45

**Query Syntax**

[:SENSe]: AVERage:COUNt?

**Return Type**
Numeric

**Default**
SA - Depends on measurement
IQA - 10

---

Last Modified:

22-Oct-2017 Added IQA mode (10.1x)
24-Mar-2014 Added

---

[:SENSe]:AVERage[:ENABle] <value>

(Read-Write) Set and query the averaging state.

**Relevant Modes**
CPM

**Parameters**

<value> (Boolean) Choose from:

ON (or 1) - Averaging ON. Set the number of averages using [:SENSe]:AVERage:COUNt
OFF (or 0) - Performs NO averaging.

**Examples**

AVER 1

**Query Syntax**

[:SENSe]:AVERage[:ENABle]?

**Return Type**
Boolean

**Default**
ON

---

Last Modified:

24-Mar-2014 New command

---

[:SENSe]:AVERage[:MODE] <char>
(Read-Write) Set and query the averaging mode for Power Meter measurements.

**Relevant Modes** Power Meter, Pulse Measurements

**Parameters**

- `<char>` Average Mode.
  - **AUTO** - Allows the USB Power Sensor to use its Auto Averaging settings.
  - **MAN** - Manually enter an averaging count by setting `SENSe:AVERage:COUNt`.
  - **OFF** - Performs NO averaging.

**Examples** AVER MAN

**Query Syntax** [:SENSe]:AVERage[:MODE]?

**Return Type** Character

**Default** AUTO

[:SENSe]:AVERage:MODE <char>

(Read-Write) Set and query the averaging mode for NA measurements.

**Relevant Modes** NA

Only Sweep averaging is allowed in CAT mode.

**Parameters**

- `<char>` Average Mode. Choose from:
  - **SWEep** - Each data point is based on the average of the same data point being measured over `<n>` consecutive sweeps. The average counter shows the number of previous sweeps that have been averaged together to form the current trace. When the counter reaches the specified count, then a ‘running average’ of the last `<n>` sweeps is displayed.
  - **POINT** - Each data point is measured `<n>` times and averaged before going to the next data point. On subsequent sweeps, averaging restarts by measuring each data point again `<n>` times. The average counter is not updated because data is not displayed until all the averages have been applied.

**Examples** AVER:MODE POINT

**Query Syntax** [:SENSe]:AVERage:MODE?

**Return Type** Character

**Default** SWEEP

---

Last Modified:

17-Jul-2012 New command (6.00)
(Read-Write) Set and query the step detection mode state for Power Meter measurements.

**Relevant Modes**  
Power Meter

**Parameters**

<bool>  
Step Detection state. Choose from:
- **ON** (or 1) - Step Detection enabled.
- **OFF** (or 0) - Step Detection disabled.

**Examples**  
AVER:SDET On

**Query Syntax**  
[:SENSe]:AVERage:SDETect?

**Return Type**  
Boolean

**Default**  
ON

### [:SENSe]:AVERage:TYPE <char>

(Read-Write) Set and query the averaging type for IQA and SA measurements.

**Relevant Modes**  
IQA, SA

**Parameters**

<char>  
Average type. Choose from:
- **AUTO** - (SA Only) The FieldFox chooses the most appropriate type of averaging for the current settings. When Detection Method is set to Average, a Noise Marker is present, or a Channel measurement is active, then Power Average is ALWAYS selected. Otherwise, Log Average is selected.
- **POWer** - (Linear) Averaging – Best for measuring true power levels. Used in Detection Average and Noise Marker Average. Mathematically, trace noise is 2.5 dB higher than when using Log Average. PAvg is shown on the left side of the FieldFox screen when selected.
- **LOG** - Best for displaying Trace Averaging. LgAv is shown on the left side of the FieldFox screen when selected.
- **VOLT** - Best for measuring true voltage levels.

**Examples**  
AVER:TYPE POW

**Query Syntax**  
[:SENSe]:AVERage:TYPE?

**Return Type**  
Character

**Default**  
AUTO

---

Last Modified:  
22sept2017  
Added IQA mode (10.1x)
(Read-Write) Set and query IF Output state.

**Relevant Modes**
SA Mode
All models with an IF Output connector.

**Parameters**

<char> Choose from:
- **OFF** - The IF output signal is NOT processed.
- **NARRow** - The IF output signal is processed and has approximately 5 MHz bandwidth.
- **WIDE** - The IF output signal is processed and has approximately 25 MHz bandwidth.

**Examples**

```
BAND:IF:OUT NARR
```

**Query Syntax**

[:SENSe]:BANDwidth:IF:OUT?

**Return Type**
Character

**Default**
OFF

---

Last modified:
17-Jul-2012 New command (A.06.00)

[:SENSe]:BANDwidth[:RESolution] <num>

(Read-Write) Set and query the resolution bandwidth.

**Relevant Modes**
SA, RTSA

**Parameters**

<num> Resolution Bandwidth in Hz. Choose from: 10 Hz to 2 MHz.
This command will accept MIN and MAX as arguments.

**Examples**

```
BAND 10e3
```

**Query Syntax**

[:SENSe]:BANDwidth[:RESolution]?

**Return Type**
Numeric

**Default**
2 MHz

---

Last Modified:
19-sep-2016 Added RTSA to relevant modes.

[:SENSe]:BANDwidth[:RESolution]:AUTO <bool>

(Read-Write) Set and query the automatic resolution bandwidth state.

**Relevant Modes**
SA, RTSA
Parameters

<bool> Choose from:

O or OFF - Set Resolution BW manually using \texttt{BAND:RES <num>}

1 or ON - Automatic Bandwidth setting

Examples \texttt{BAND: AUTO 0}

Query Syntax \texttt{[:SENSe]:BANDwidth[:RESolution]:AUTO?}

Return Type Boolean

Default ON

Last Modified:

19-sep-2016 Added RTSA to relevant modes.

[:SENSe]:BWIDTH:VIDeo <char>

\textit{(Read-Write)} Set and query the video bandwidth for the measurement.

Relevant Modes \textit{Pulse Measurements}

Parameters

<char> Video Bandwidth. Choose from the following:

- \texttt{OFF} - Video Bandwidth is disabled.
- \texttt{LOW} - Similar to a low pass filter, ripple in the pass band is minimized but allows higher side-lobes on the filter skirt.
- \texttt{MEDIUM} - Smooth pass band with reasonable transition ripple.
- \texttt{HIGH} - The pass band ripple is similar to the OFF setting, but the transition skirts are smoother.

Examples \texttt{BWID:VID LOW}

Query Syntax \texttt{[:SENSe]:BWIDTH:VIDeo?}

Return Type Character

Default OFF

Last modified:

31-Oct-2013 New command

[:SENSe]:BANDwidth:VIDeo <num>

\textit{(Read-Write)} Set and query the video bandwidth. Also set \texttt{BAND:VID:AUTO OFF}.

Relevant Modes SA
Parameters

<i>num</i>  Video bandwidth. Choose a value between 1 and 2E6.

**Examples**

```
BAND:VID 1e3
```

**Query Syntax**

```
SENSe:BANDwidth:VIDeo?
```

**Return Type**

Numeric

**Default**

2E6

[SENSe]:BANDwidth:VIDeo:AUTO <bool>

*(Read-Write)* Set and query the ON | OFF state of Video Bandwidth.

**Relevant Modes**

SA

**Parameters**

<i>bool</i>  Choose from:

- **O** or **OFF**  Set Video BW manually using BAND:VIDeo <num>
- **1** or **ON**  Automatic Bandwidth setting.

**Examples**

```
BAND:VID:AUTO 1
```

**Query Syntax**

```
:SENSe:BANDwidth:VIDeo:AUTO?
```

**Return Type**

Boolean

**Default**

ON

[SENSe]:BURSt:ALIgnment:NOW

This command has been replaced by [:SENSe]:ALIgnment:BURSt[:STATe]. Learn about superseded commands.

*(Write-only)* A RF Burst alignment is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

**Relevant Modes**

IQA, SA & RTSA Modes

All models with RF Burst triggering (All EXCEPT N9912A)

**Parameters**

None

**Examples**

```
BURS:ALIG:NOW
```

**Query Syntax**

Not Applicable

**Default**

Not Applicable

---

Last Modified:

22nov2017  Updated as a superseded note & added IQA for backwards compatibility.

20-sep-2016  Add RTSA mode (A.09.50)
[SENSe]:BWIDth <num>

(Read-Write) Set and query the IF bandwidth for the measurement.

**Relevant Modes**  
Pulse Measurements, NA

**Parameters**

<num>  
IF Bandwidth.

Choices: 10 | 30 | 100 | 300 | 1000 | 10000 | 30000 | 100000

Minimum: 10 Hz  
Maximum: 100 kHz

**Examples**

BWID 30  
BWID 10e4

**Query Syntax**  
[:SENSe]:BWIDth?

**Return Type**  
numeric

**Default**  
10000

---

Last modified:

31-Oct-2013  New command

---

[SENSe]:CMEasurement:AVERage:ENABle <bool>

(Read-Write) Set and query the state of averaging for an SA channel measurement.

**Relevant Modes**  
SA

**Parameters**

<bool>  
Average State. Choose from:

OFF or 0 - Averaging OFF  
ON or 1 - Averaging ON

**Examples**

CME:AVER:ENAB 1

**Query Syntax**  
[:SENSe]:CMEasurement:AVERage:ENABle?

**Return Type**  
Boolean

**Default**  
ON

---

[SENSe]:CMEasurement:IBW <num>
**Command Reference**

(Read-Write) Set and query the Channel Integrating Bandwidth. This value specifies the range of integration used in calculating the power in the main channel. It is applied to both ACP and Channel Power Measurements in SA mode.

**Relevant Modes** SA

**Parameters**

$num$ Channel Integrating Bandwidth value in Hz. Choose a number between the Minimum Integration Bandwidth (100 Hz) and Maximum Integration Bandwidth (3 GHz).

**Examples** SENS:CME:IBW 3e6

**Query Syntax** [:SENSe]:CMEasurement:IBW?

**Return Type** Numeric

**Default** 2 MHz

[:SENSe]:CMEasurement:RRCFilter $bool$

(Read-Write) Set and query the state of RRC (Root-Raised-Cosine) filter weighting for Channel Power and ACPR measurements.

**Relevant Modes** SA

**Parameters**

$bool$ RRC State. Choose from:
- **OFF** or 0 - RRC weighting OFF
- **ON** or 1 - RRC weighting ON

**Examples** CME:RRCF 1

**Query Syntax** [:SENSe]:CMEasurement:RRCFilter?

**Return Type** Boolean

**Default** OFF

[:SENSe]:CMEasurement:RRCFilter:ALPHa $num$

(Read-Write) Set and query the value of RRC (Root-Raised-Cosine) filter weighting for Channel Power and ACPR measurements. When RRC weighting is applied to transmitted and received power, the edges of the channel are "smoothed" to help prevent interference.

**Relevant Modes** SA

**Parameters**

$num$ RRC value. Choose a value between 0 (no smoothing) and 1 (most smoothing).

**Examples** CME:RRCF:ALPH .50
Query Syntax  [:SENSe]:CMEasurement:RRCFilter:ALPHA?
Return Type  Numeric
Default  .22

[SENSe]:CORRection:LOSS:AFTer:TEMPerature <double>
(Read-Write) Set and query the loss correction values after the DUT (DUT In). Returns a the temperature correction in double numerical format.
See also CORRection:LOSS:AFTer[:VALue] and CORRection:LOSS:AFTer:ENABLEd.
Relevant Modes  NF
Parameters
<double numbers>  Enter values in Centigrade, Fahrenheit, or Kelvin. All values converted to Kelvin.
Examples  :CORR:LOSS:AFT:TEMP 87.43 (Converts any value entered to Kelvin units.)
Query Syntax  :CORRection:LOSS:AFTer:TEMPerature?
Return Type  double numbers
Default  296.50K

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)

[SENSe]:CORRection:LOSS:BEFore:TEMPerature <double>
(Read-Write) Set and query the loss correction values before the DUT (DUT In). Returns a the temperature correction in double numerical format.
See also CORRection:LOSS:BEFore[:VALue] and CORRection:LOSS:BEFore:ENABLEd.
Relevant Modes  NF
Parameters
<double numbers>  Enter values in Centigrade, Fahrenheit, or Kelvin. All values converted to Kelvin.
Examples  :CORR:LOSS:BEF:TEMP 87.43 (Converts any Centigrade or Fahrenheit value entered to Kelvin units.)
Query Syntax  :CORRection:LOSS:BEF:TEMPerature?
Return Type  double numbers
Default  296.50K
[:SENSe]:CORRection:CALReady:TYPE <char>

(Write-Read) Specifies the type of CalReady calibrations that is performed when the FieldFox is Preset.

**Relevant Modes** CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- <char>
  - Choose from:
    - **TWOPort** - Corrects all four S-parameters. Requires a forward and reverse sweep, which causes slower trace measurements.
    - **ERESponse** - (Enhanced response) Corrects forward (S21 and S11) and reverse (S12 and S22) measurements separately. Therefore, when measurements in only one direction are required, this choice provides faster trace measurements than a full 2-port cal.

**Note:** There is no programming command to choose 'reciprocal' for Enhanced Response as there is from the User Interface.

**Examples**

```
CORR:COLL:CALR:TYPE TWOP
```

**Query Syntax** [:SENSe]:CORRection:CALReady:TYPE?

**Default** TWOPort

---

[:SENSe]:CORRection:COAX <char>

(Read-Write) Set and query the state of Cable Correction in DTF measurements.

**Relevant Modes** CAT

**Parameters**

- <char>
  - Choose from:
    - **MAN** - DTF cable specifications are entered manually.
    - **AUTO** - DTF cable specifications are entered from a cable file.

**Examples**

```
CORR:COAX MAN
```

**Query Syntax** [:SENSe]:CORRection:COAX?

**Return Type** Character
Default  MAN

[:SENSe]:CORRection:COEFFicient[:DATA] <char>,<resp1,stim2>,<data>

(Read-Write) Set and query the correction coefficients (error terms) for the current calibration.

**Relevant Modes**  CAT, NA

**Parameters**

- **<char>** Error term. Choose from:
  - **ES** - Source match
  - **ER** - Reflection tracking
  - **ED** - Directivity
  - **All models EXCEPT N9912A:**
    - **EL** - Load Match
    - **ET** - Transmission tracking
    - **EX** - Isolation

- **<resp1,stim2>** Response port, Stimulus port.
  For N9912A, choose 1,1.
  For all other models, choose from 1 or 2 for each port.
  - For ES, ER, or ED, the response port and the stimulus port must be the same.
  - For EL, ET, or EX, the response port and the stimulus port must be different.

- **<data>** Two values per data point: (Real value, Imaginary value), separated by commas.

**Examples**

'Writes 3 data points of complex source match data
CORR:COEF ES,1,1,2.626292E-01,1.823261E-01,2.537209E-01,1.943861E-01,2.448751E-01,2.059776E-01

**Query Syntax**  [:SENSe]: CORRection:COEFFicient[:DATA] <char>, <resp1, stim2>

**Return Type**  Numeric

**Default**  Not Applicable

---

Last Modified:

18-Oct-2012  Added new models

---

[:SENSe]:CORRection:COLLect[:ACQuire]:INT <num>

(WRITE-ONLY) Measures the internal (Quick Cal) standard for the specified port. Prompt to leave the port OPEN for this step.

Use [:SENSe]:CORRection:COLLect:METHod:QCALibrate:CALibrate for both 1-port and 2-port QuickCal.
Use `[:SENSe]:CORRection:COLLect[:ACQuire]:LOAD` for the optional LOAD measurement.

This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with `*OPC?` to cause the FieldFox to wait before accepting subsequent commands.

### Relevant Modes
- CAT, NA
- To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

### Parameters
- `<num>` Port number to calibrate.
  - N9912A - Choose 1 (RF OUT)
  - All others - Choose 1 or 2

### Examples
- `CORR:COLL:INT 1`
- `See Cal Examples`

### Query Syntax
- Not Applicable

### Default
- Not Applicable

---

Last Modified: 18-Oct-2012 Modified for new models

---

Use `[:SENSe]:CORRection:COLLect[:ACQuire]:LOAD <num>`

(Write-Only) Measures the LOAD calibration standard that is connected to the specified port.

This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with `*OPC?` to cause the FieldFox to wait before accepting subsequent commands.

### Relevant Modes
- CAT, NA
- To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

### Parameters
- `<num>` Port number to which the LOAD standard is connected.
  - N9912A - Choose 1 (RF OUT)
  - All other models - Choose 1 or 2

### Examples
- `CORR:COLL:LOAD 1`
- `See Cal Examples`

### Query Syntax
- Not Applicable

### Default
- Not Applicable

---

Last Modified:
### [:SENSe]:CORRection:COLLect[:ACQuire]:OPEN <num>

*(Write-Only)* Measures the OPEN calibration standard that is connected to the specified port. This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with *OPC?* to cause the FieldFox to wait before accepting subsequent commands.

**Relevant Modes**  
CAT, NA  
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- `<num>` Port number to which the OPEN standard is connected.
  - N9912A - Choose 1 (RF OUT)
  - All other models - Choose 1 or 2

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:COLL:OPEN 1</td>
<td>See Cal Examples</td>
</tr>
</tbody>
</table>

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

### [:SENSe]:CORRection:COLLect[:ACQuire]:SHORt <num>

*(Write-Only)* Measures the SHORT calibration standard that is connected to the specified port. This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with *OPC?* to cause the FieldFox to wait before accepting subsequent commands.

**Relevant Modes**  
CAT, NA  
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- `<num>` Port number to which the SHORT standard is connected.
  - N9912A - Choose 1 (RF OUT)
  - All other models - Choose 1 or 2

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:COLL:SHOR 1</td>
<td>See Cal Examples</td>
</tr>
</tbody>
</table>

---

Last Modified:  
18-Oct-2012 Added new models
[:SENSe]:CORRrection:COLLect[:ACQuire]:THRU <p1>,<p2>

(Write-Only) Measures the THRU calibration standard that is connected between the specified ports. This can be used in any of the following Cal Methods:

- **THRU Response** - both reference planes are connected.
- **Unknown THRU** - any cable or adapter is used to connect both reference planes.
- **Enhanced Response** - both reference planes are connected.
- **Quick Cal Enhanced Response** - both reference planes are connected.

This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with "*OPC?" to cause the FieldFox to wait before accepting subsequent commands.

**Relevant Modes**  CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

<p1>,<p2>  Port numbers to which the THRU standard is connected.

Choose 1,2

**Examples**  CORR:COLL:THRU 1,2

See Cal Examples

**Query Syntax**  Not Applicable

**Default**  Not Applicable

Last Modified:

18-Oct-2012  Reviewed for new models

[:SENSe]:CORRrection:COLLect:CKIT:LABel <port>,<ckit>

(Write-Read) Set and read the Cal Kit to use for the specified port number.

**Relevant Modes**  CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**
<port> Port number for the Cal Kit.
   N9912A, choose 1
   All other models, choose 1 or 2

<ckit> (String) Cal Kit for the specified port number. Case-sensitive.
   Use \[:SENSe:]CORR:COLL:CKIT:LAB:CATalog? to read a list of valid Cal Kits.
   CORR:COLL:CKIT:LAB 1, "85052D"
   See Cal examples

Using ECal

ECal modules that are connected to a FieldFox USB port with connectors that match the specified <port> are appended to the end of the list of valid Cal Kits that are returned using \[:SENSe:]CORR:COLL:CKIT:LAB:CATalog?.

NOTE: The ECal module MUST be connected.

To specify an ECal module, instead of <ckit> use the following (String) syntax: <model>:<serial>:<char index>
   • <model> - ECal module model number
   • <serial> - Optional argument. If unspecified, then the first module detected with a matching model and characterization index will be used.
   • <char index> - Optional argument. If unspecified, then 0 (factory characterization) is used.

The following are valid ECal specifiers:

'Specify all three arguments including User Char 1:
   CORR:COLL:CKIT:LAB 1, "N4431A;02673;1"
'The first N4431A will be used with the Factory Characterization
   CORR:COLL:CKIT:LAB 1, "N4431A"
'Use the Factory Char.
   CORR:COLL:CKIT:LAB 1, "N4431A;02673"
'The first N4431A will be used with User Char 1
   CORR:COLL:CKIT:LAB 1, "N4431A;;1"
   See ECal example program

Query Syntax \[:SENSe:]CORR:COLL:CKIT:LAB? <port>

Default Not Applicable

Last Modified:
26-Jul-2013 Added ECal
18-Oct-2012 Added new models
[:SENSe]:CORRection:COLLect:CKIT:LABel:CATalog?

(Read-Only) Reads the available Cal Kits in the FieldFox.

ECal reference strings for attached ECal modules are included in the list. The format of ECal reference strings can be found at [:SENSe]:CORRection:COLLect:CKIT:LABel.

**Relevant Modes**  CAT, NA

**Parameters**  None

**Examples**  
CORR:COLL:CKIT:LAB:CAT?

'Returns comma-separated strings:
'The following is NOT a complete list
"85054D","85052D","85039B","85038A","85036B/E", "85033D/E","85032F","85032B/E","85031B"

See Cal examples

**Default**  Not Applicable

Last Modified:

26-Jul-2013  Edited for ECal

18-Oct-2012  Reviewed for new models

[:SENSe]:CORRection:COLLect:CONNector <port>,<type -gen-,imp>

(Write-Read) Set and read the DUT connector that will be connected to the specified FieldFox port.

**Relevant Modes**  CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- **<port>**  Port number for the Cal Kit.
  N9912A, choose 1
  All other models, choose 1 or 2

- **<type -gen-,imp>**  (String) Type, gender, and impedance of the DUT connector.
  > Type: choose from: Type N | 3.5 mm | 7 mm | 7/16 | Type F | WR-10 | WR-15 | WR-19 | WR-22 | WR-28 | WR-42 | WR-62 | WR-75 | WR-90 | WR-112 | WR-137 | WR-187 | WR-284 | WR-650 | Other
  > Gender: choose from one of the following when the connector type includes gender. 7mm, 7/16, and waveguide (WR-<nn>) do NOT include gender.
    - -M-  (Male)
    - -F-  (Female)
    - Be VERY careful with the leading and trailing dashes. Some text editors change the dash style.
- Provide a space before each.
  > **Impedance**: choose from: **50** or **75**.
- Provide a leading comma as in the following examples.

A regular expression/compact sequence for the connector name could be expressed as:
"[^\s].* (\-M|F-)?,\d+"

### Examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:COLL:CONN 1,&quot;Type N -M-50&quot;</td>
<td>Correction:Collect:Connector 1, &quot;Type N -M-50&quot;</td>
</tr>
<tr>
<td>Correction:Collect:Connector 2, &quot;7 mm,50&quot;</td>
<td></td>
</tr>
<tr>
<td>CORR:COLL:CONN 1, &quot;WR-10,1&quot;</td>
<td></td>
</tr>
<tr>
<td>CORR:COLL:CONN 2, &quot;WR-650,1&quot;</td>
<td></td>
</tr>
</tbody>
</table>

See Cal examples

### Query Syntax

```
[:SENSe]:CORRection:COLLect:CONNector? <port>
```

**Default** Not Applicable

---

### Last Modified:

- 22-Oct-2012 Updated for new models
- 16-Feb-2011 Updated

```
[:SENSe]:CORRection:COLLect:ECAL:AORient <bool>
```

(Write-Read) Set and read the state of auto orientation for a calibration using an ECal module.

**Relevant Modes**

CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- `<bool>`: Auto orientation state. Choose from the following:
  - **ON** or 1: Automatically detect the orientation of the ECal module.
  - **OFF** or 0: Connect the ECal module to the FieldFox ports as prompted during calibration. Select when using low power to the module.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:COLL:ECAL:AOR 1</td>
<td></td>
</tr>
<tr>
<td>Sense:Correction:Collect:ECAL:Aorient off</td>
<td>See Cal examples</td>
</tr>
</tbody>
</table>

### Query Syntax

```
[:SENSe]:CORRection:COLLect:ECAL:AORient ?
```

**Default** ON or 1

---

**Last Modified:**

- 26-Jul-2013 New topic

---

359
[:SENSe]:CORRection:COLLect:ECAL:SIMPle <bool>

(Write-Read) Choose to receive a single prompt to connect the ECal or one for each reference plane.

Note: This command should be sent before sending any of the SENS:CORR:COLL:METH commands. See Calibration Commands.

Relevant Modes
CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters
<bool> Simple ECal state. Choose from the following:
ON or 1 (default) - The FieldFox assumes that both ports of the ECal module can be connected at both calibration reference planes. You therefore receive a single prompt to connect the ECal module to both ports.
OFF or 0 - Select when physical restrictions prohibit the ECal module from being connected to both calibration reference planes simultaneously. You will receive separate prompts to connect the ECal module to Port 1, then to connect the ECal to Port 2.

Examples
CORR:COLL:ECAL:SIMP
Sense:Correction:Collect:Ecal:Simple off
See Cal examples

Query Syntax
[:SENSe]:CORRection:COLLect:ECAL:SIMPle ?

Default
ON or 1

Last Modified:
1-Nov-2013 New command

[:SENSe]:CORRection:COLLect:GUIDed:SCOunt?

(Read-Only) Returns the number of steps required to complete the guided calibration.

Relevant Modes
CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters
None

Examples
CORR:COLL:GUID:SCO?
See Example Program

Default
Not Applicable

Last Modified:
18-Oct-2012 New command
[:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire <num>

(Write-Only) Measures the standard for the specified step number in the guided cal. Use [:SENSe]:CORRection:COLLect:GUIDed:SCOunt to read the number of steps required for the cal. This command is overlapped. This means that it will execute and, while the FieldFox measures the cal standard, it will also accept new commands. Follow this command with *OPC? to cause the FieldFox to wait before accepting subsequent commands.

Relevant Modes
CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters

<nnum> Step number of the guided calibration.

Examples
CORR:COLL:GUID:STEP:ACQ 1
See Guided Cal Example

Query Syntax
Not Applicable

Default
Not Applicable

Last Modified:
18-Oct-2012 New command

[:SENSe]:CORRection:COLLect:GUIDed:STEP:PRoMpt? <num>

(Read-Only) Returns the cal standard prompt to be used for the specified step number in the guided cal. Use [:SENSe]:CORRection:COLLect:GUIDed:SCOunt to read the number of steps required for the cal. Use [:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire to perform the cal std measurement.

Relevant Modes
CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters

<nnum> Step number of the guided calibration.

Examples
CORR:COLL:GUID:STEP:PRM? 1
See Guided Cal Example

Query Syntax
Not Applicable

Default
Not Applicable

Last Modified:
18-Oct-2012 New command
[:SENSe]:CORRection:COLLect:METHod:ERES <p1>,<p2>

(Write-Only) Sets the Cal method to Enhanced Response. This Cal Method requires mating (insertable) DUT connectors. Use [:SENSe]:CORRection:COLLect:CONNector to change connector and gender.

Prompt for, then measure standards:
- OPEN, SHORT, and LOAD on the <p1> (source) port.
- THRU connection between the ports.

Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes  CAT, NA

Parameters
<p1>,<p2>  Port numbers to calibrate.
For N9912A with Opt 110, choose 1,2.
For all other models with Full S-parameter option, choose 1,2 (forward) or 2,1 (reverse).

Examples  CORR:COLL:METH:ERES 1,2
See Cal examples

Query Syntax  Not Applicable

Default  Not Applicable

Last Modified:
18-Oct-2012  Added new models

[:SENSe]:CORRection:COLLect:METHod:ORESponse <p1>

(Write-Only) During a Guided calibration, sets the Cal method to 1-port Response using an Open standard.

Prompt for, then measure standards:
- Open on the <p1> port.

Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes  CAT, NA (NOT available on N9912A)
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters
<p1>  Port number to be calibrated.

Examples  CORR:COLL:METH:ORES 2
See Cal examples

Query Syntax  Not Applicable
[:SENSe]:CORRrection:COLLect:METHod:QCALibrate:CALibrate <p1>[,p2]

(Write-Only) Sets the Cal Method to 1-port or 2-port QuickCal.

For 1-port QuickCal:
- Prompt for INT Stds on <p1>, then optional LOAD on <p1>

For 2-port QuickCal with Insertable connectors, performs QSOLT.
- Prompt for INT Stds on <p1>, then optional LOAD on <p1>, then Flush THRU

For 2-port QuickCal with Non-Insertable connectors, performs SOLR.
- Prompt for INT Stds on <p1> and <p2>, then optional LOAD on <p1> and <p2>, then ANY THRU.

Learn more about FieldFox Cal Methods using SCPI.

**Relevant Modes**
CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**
<p1> First port to calibrate.
- For N9912A, choose 1.
- For all other models, choose 1 or 2.

[p2] For all models with Full S-parameter option, second port to calibrate.
Choose 1 or 2 (alternate to <p1>).

**Examples**
CORR:COLL:METH:QCAL:CAL 1,2
See Cal examples

**Query Syntax**
Not Applicable

**Default**
Not Applicable

Last Modified:
18-Oct-2012 Added new models
or reverse (S22/S12) measurements.
This Cal Method requires mating (insertable) DUT connectors. Use [:SENSe]:CORRection:COLLect:CONNector to change connector and gender.
Requires QuickCal Option.
- Prompt to "Leave ports OPEN", then use [:SENSe]:CORRection:COLLect:[ACQuire]:INT to measure internal OPEN and SHORT.
- Optional - Prompt to "Connect LOAD to ports", then use [:SENSe]:CORRection:COLLect:[ACQuire]:LOAD to measure LOAD.
- For 2-port Cal - Prompt to connect THRU between ports, then use [:SENSe]:CORRection:COLLect:[ACQuire]:THRU to measure THRU
Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes N9912A with Opt 110: CAT, NA - ONLY <1,2> are supported (NOT 2,1 reverse)
All other models with Full S-parameter option, CAT, NA modes.
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters</p1>
First port.
- For N9912A, choose 1.
- For all other models, choose 1 or 2.
<p2>
Second port to calibrate.
- For N9912A, choose 2.
- For all other models, choose 1 or 2 (alternate to <p1>).

Examples
CORR:COLL:METH:QCAL:ERES 1,2

See Cal example

Query Syntax Not Applicable

Default Not Applicable

Last Modified:
18-Oct-2012 Added new models

[:SENSe]:CORRection:COLLect:METHod:QSOLt <p1,p2>

(Write-Only) Sets the Cal Method to 2-port QSOLT. Calibrate for sweeps in BOTH directions.
This Cal Method is NOT available on N9912A and all other models without full 2-port S-parameter option.
This Cal Method requires mating (insertable) DUT connectors. Use [:SENSe]:CORRection:COLLect:CONNector to change connector and gender.
Prompt for, then measure standards:
- OPEN, SHORT, and LOAD on the <p1> (source) port. (This selection is NOT available from the user interface.)
• FLUSH THRU connection between the ports.
This Cal Method requires mating (insertable) connectors. Use [:SENSe]:CORRection:COLLect:CONNECTor to change connector and gender.
Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes  CAT and NA Mode
               To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters
<p1,p2> Port numbers to calibrate.
Choose 1,2 or 2,1

Examples  CORR:COLL:METH:QSOLt 1,2
See Cal example

Query Syntax  Not Applicable
Default  Not Applicable

Last Modified:
18-Oct-2012 Added new models

[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:OPEN <p1>

(Write-Only) Sets the Cal method to Open Response using a Mechanical Cal Kit.
Prompt for, then measure standards:
• OPEN on the <p1> port.
Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes  CAT, NA
               To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters
<p1> Port number to be calibrated.
For N9912A and all other models without full 2-port S-parameter option, choose 1.
For all models with full 2-port S-parameter option, choose 1 or 2.

Examples  CORR:COLL:METH:OPEN 2
See Cal examples

Query Syntax  Not Applicable
Default  Not Applicable

Last Modified:
[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:SHORt <p1>

(Write-Only) Sets the Cal method to Short Response using a Mechanical Cal Kit.
Prompt for, then measure standards:
- **SHORT** on the <p1> port.
- **THRU** connection between the ports.

Learn more about FieldFox Cal Methods using SCPI.

**Relevant Modes**
- CAT, NA
  - To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**
- **<p1>** Port number to be calibrated.
  - For N9912A and all other models without full 2-port S-parameter option, choose 1.
  - For all models with full 2-port S-parameter option, choose 1 or 2.

**Examples**
- **CORR:COLL:METH:SHOR 1**

**Query Syntax**
- Not Applicable

**Default**
- Not Applicable

---

Last Modified:
18-Oct-2012 Edited for new models

[:SENSe]:CORRection:COLLect:METHod[:RESPonse]:THRU <ports>

(Write-Only) Sets the Cal method to THRU Response (also known as Normalization cal).
Prompt for, then measure standard:
- **THRU** connection between the ports.

Learn more about FieldFox Cal Methods using SCPI.

**Relevant Modes**
- CAT, NA
  - To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**
- **<ports>** Port numbers to be calibrated. Choose 1,2
[:SENSe]:CORRection:COLLect:METHod:SOLT1 <p1>

(Write-Only) Sets the Cal Method to 1-port OSL calibration. Prompt for, then measure standards:

- OPEN, SHORT, and LOAD on the <p1> (source) port.

Learn more about FieldFox Cal Methods using SCPI.

Relevant Modes

- CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters

- <p1> Port number to be calibrated.
  - For N9912A, choose 1.
  - For all other models, choose 1 or 2.

Examples

CORR:COLL:METH:THRU 1,2

See Cal examples

Query Syntax

Not Applicable

Default

Not Applicable

Last Modified:

18-Oct-2012 Edited for new models

[:SENSe]:CORRection:COLLect:METHod:SOLT2 <p1,p2>

(Write-Only) Sets the Cal Method to 2-port SOLT calibration. Calibrate for sweeps in BOTH directions. This Cal Method is NOT available on N9912A and all other models without Opt 122.

NOTE: [:SENSe]:CORRection:COLLect:METHod:SOLR is usually a more accurate 2-port calibration. Learn more about FieldFox Cal Methods using SCPI.

Prompt for, then measure standards:

- OPEN, SHORT, and LOAD on BOTH ports.

Last Modified:

18-Oct-2012 Edited for new models
• **THRU** connection between the ports. This should be a known (characterized) THRU standard.

**Relevant Modes** CAT and NA Mode

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- <p1>,<p2> Port numbers to be calibrated. Choose 1,2.

**Examples**

- `CORR:COLL:METH:SOLT2 1,2`

*See Cal examples*

**Query Syntax** Not Applicable

**Default** Not Applicable

---

[:SENSe]:CORRrection:COLLect:METHod:SOLR <p1>,<p2>

(Write-Only) Sets the Cal Method to Short-Open-Load-Reciprocal Thru. (Also known as 2-port Unknown Thru).

This Cal Method is NOT available on the N9912A.

Prompt for, then measure standards:

- **OPEN**, **SHORT**, and **LOAD** on the <p1> and <p2> source ports.
- **THRU** connection between the ports.

**The Unknown Thru Standard:**

- Can have up to about 40 dB of loss and long electrical length.
- Must be reciprocal: S21=S12.

*Learn more about FieldFox Cal Methods using SCPI.*

**Relevant Modes** CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

- <p1>,<p2> Port numbers to be calibrated. Choose 1,2

**Examples**

- `CORR:COLL:METH:SOLR 1,2`

*See Cal examples*

**Query Syntax** Not Applicable

**Default** Not Applicable

---

Last Modified: 18-Oct-2012 Edited for new models
[:SENSe]:CORRection:COLLect:METHod:SRESponse <p1>

(Write-Only) During a Guided calibration, sets the Cal method to 1-port Response using a Short standard. Prompt for, then measure standards:
- **SHORT** on the <p1> port.

Learn more about FieldFox Cal Methods using SCPI.

**Relevant Modes**  
CAT, NA (NOT available on N9912A)
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

| <p1> | Port number to be calibrated. |

**Examples**

CORR:COLL:METH:SRES 2  
See Cal examples

**Query Syntax**  
Not Applicable

**Default**  
Not Applicable

---

[:SENSe]:CORRection:COLLect:METHod:TRL <p1,p2>

(Write-Only) Sets the Cal Method to 2-port TRL calibration. Calibrate for sweeps in BOTH directions. This Cal Method is NOT available on N9912A.

Use the **Guided Calibration interface** to:

1. Select the DUT connectors: [:SENSe]:CORRection:COLLect:CONNector
2. Select a TRL Cal Kit: [:SENSe]:CORRection:COLLect:CKIT:LABel
3. Query number of steps: [:SENSe]:CORRection:COLLect:GUIDed:SCOunt
5. Acquire standards: [:SENSe]:CORRection:COLLect:GUIDed:STEP:ACQuire

**Relevant Modes**  
CAT and NA Mode  
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

| <p1,p2> | Port numbers to be calibrated. Choose 1,2. |
[:SENSe]:CORRection:COLLect:METHod:TYPE?

(Read-Only) Query the current calibration type.

**Relevant Modes**
CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

**Examples**
CORR:COLL:METH:TYPE?
See Cal examples

**Default**
Depends on installed options.

Last Modified:
18-Oct-2012 Edited for new models

[:SENSe]:CORRection:COLLect:OISolation <bool>

(Write-Read) Set and query the state of the Omit Isolation setting.

The optional isolation step of a calibration corrects for crosstalk which is the internal signal leakage between the test ports. The additional Isolation step measures Load standards that are connected to both FieldFox test ports.

Perform an isolation calibration when you are testing a device with high insertion loss, such as the stop band of a filter or a switch in the open position. See FieldFox User’s Guide for more information.

**Relevant Modes**
CAT, NA

To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

**Parameters**

<bool> Choose from the following:
ON or 1 - Omit the Isolation step.
OFF or 0 - Perform the Isolation step
Examples

CORR:COLL:OIS 1
Sense:Correction:Collect:OIsolation off

Query Syntax

[:SENSe]:CORRection:COLLect:OISolation?

Default

ON or 1

Last Modified:

1-Nov-2013 New command

[:SENSe]:CORRection:COLLect:SAVE <num>

(Write-Only) Ends the calibration, computes the error terms, and turns correction ON.

Relevant Modes

CAT, NA
To Cal in VVM mode, perform a Cal in CAT or NA mode, then switch to VVM mode.

Parameters

<num> Cal Set to which the Cal is saved. Choose 0

Examples

CORR:COLL:SAVE 0
See Cal Examples

Query Syntax

Not Applicable

Default

Not Applicable

Last Modified:

18-Oct-2012 Reviewed for new models

[:SENSe]:CORRection:DUT:MODe <char>

(Read-Write) Set and query the DUT uncertainty setup values being measured.

See also [:SENSe]:CORRection:DUT:INGamma, [:SENSe]:CORRection:DUT:OUTGamma, [:SENSe]:CORRection:DUT:SPEC, [:SENSe]:CORRection:DUT:DISTribution.
See also, MMEMory:LOAD:DUT and MMEMory:STORe:DUT.

Relevant Modes

NF

Parameters

<character> DUT mode uncertainty settings. Choose from:

SPOT - Sets the DUT mode to Spot, where you can customize the DUT uncertainty values.

TABL - Time per point can be set, but other integration SCPIs are not settable.
### Examples

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:MOD SPOT</td>
</tr>
<tr>
<td>CORR:DUT:MOD TABL</td>
</tr>
</tbody>
</table>

### Query Syntax

CORR:DUT:MOD?

### Return Type

Character

### Default

SPOT

---

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

---

## [:SENSe]:CORRRection:DUT:SPOT:DISTribution

(Read-Write) Set and query the DUT specified fixed value. This value indicates the nature of the entered value relative to the population of values it represents. (i.e., Fixed, Rayleigh, or Uniform (Uniform in Circle).

**Note:** Only available when CORR:DUT:MOD is set to SPOT.

See also [:SENSe]:CORRRection:DUT:MODe, [:SENSe]:CORRRection:DUT:SPOT:INGamma, [:SENSe]:CORRRection:DUT:SPOT:OUTGamma, [:SENSe]:CORRRection:DUT:SPOT:SPECify.

See also, MMEMory:LOAD:SPOT:DUT and MMEMory:STORe:SPOT:DUT.

### Relevant Modes

NF

### Parameters

<table>
<thead>
<tr>
<th>&lt;character&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUT Spot mode's distribution value. Choose from:</td>
</tr>
</tbody>
</table>

- **RAYL** - The probability density function (pdf) of the magnitude of the reflection coefficient is a Rayleigh distribution. This is normally the case for specified values when the overall mismatch is due to several interacting impedance transitions within the device.

- **UNIF** - A distribution that was historically used in the industry, before the Rayleigh research, as documented in Keysight Application Note 1449-3, but is rarely accurate. Refer to [https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf](https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf).

- **FIX** - The reflection coefficient magnitude is always at the given level. This is often the case when measured values are used.

### Examples

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:SPOT:DIST FIX</td>
</tr>
<tr>
<td>CORR:DUT:SPOT:DIST RAYL</td>
</tr>
</tbody>
</table>

### Query Syntax

CORR:DUT:SPOT:DIST?

### Return Type

Character

### Default

FIX

---

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)
**[:SENSe]:CORRrection:DUT:INGamma**

*(Read-Write)* Set and query the DUT's input 50Ω match, $\Gamma$ (0.000 to 1.000).

**Note:** Only available when **CORR:DUT:MOD** is set to SPOT.


See also, **MMEMory:LOAD:SPOT:DUT** and **MMEMory:STORe:SPOT:DUT**.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td>&lt;numeric&gt;</td>
</tr>
<tr>
<td>Value</td>
<td>DUT spot mode's input 50Ω match, $\Gamma$ value. Choose from: Minimum of 0.000 up to Maximum of 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:SPOT:ING .2</td>
<td>CORR:DUT:SPOT:ING 0.347</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>CORRrection:DUT:SPOT:INGamma?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

Last Modified:
01june2018        Added NF mode Opt. 356 (10.3)

**[:SENSe]:CORRrection:DUT:SPOT:OUTGamma**

*(Read-Write)* Set and query the DUT's output 50Ω match, $\Gamma$ (0.000 to 1.000).

**Note:** Only available when **CORR:DUT:MOD** is set to SPOT.


See also, **MMEMory:LOAD:SPOT:DUT** and **MMEMory:STORe:SPOT:DUT**.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td>&lt;numeric&gt;</td>
</tr>
<tr>
<td>Value</td>
<td>DUT spot mode's output 50Ω match, $\Gamma$ value. Choose from: Minimum of 0.000 up to Maximum of 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:SPOT:NGamma .2</td>
<td>CORR:DUT:SPOT:NGamma 0.347</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>CORRrection:DUT:SPOT:NGamma?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:SPOT:OUTG</td>
<td>.2</td>
</tr>
<tr>
<td>CORR:DUT:SPOT:OUTG</td>
<td>0.347</td>
</tr>
</tbody>
</table>

### Query Syntax

CORR:DUT:SPOT:OUTG?

### Return Type

Numeric

### Default

0.000E+00

---

Last Modified:

01June2018 Added NF mode Opt. 356 (10.3)

---

**[:SENSe]:CORRRection:DUT:SPOT:SPECify**

(Read-Write) Set and query the DUT specified fixed value. This value indicates the nature of the entered value relative to the population of values it represents.

**Note:** Only available when CORR:DUT:MOD is set to SPOT.

See also [:SENSe]:CORRRection:DUT:MODE, [:SENSe]:CORRRection:DUT:ISPOT:INGamma, [:SENSe]:CORRRection:DUT:SPOT:OUTGamma, [:SENSe]:CORRRection:DUT:SPOT:DIStribution.

See also, MMEMory:LOAD:DUT and MMEMory:STORe:DUT.

**Relevant Modes**  

NF

**Parameters**

- **<character>**  
  DUT Spot mode's fixed value. Choose from:

  - MAX - Use this if this value represents the largest possible reflection coefficient for the population of devices to which the DUT belongs. This is true for "does not exceed" specifications.
  - PCTL95 - Use this if 95% of the population has a lower reflection coefficient than the value entered. This is the case for "2 standard deviation" (2s) type specifications.
  - PCTL80 - Use this if 80% of the population has a lower reflection coefficient than the value entered.
  - MED or MEAN - Use this if this value represents the median / mean reflection coefficient for the population of devices to which the DUT belongs.
  - FIX - Use this if the value is a measured value for the DUT.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:DUT:SPOT:SPEC</td>
<td>PCTL95</td>
</tr>
<tr>
<td>CORR:DUT:SPOT:SPEC</td>
<td>MED</td>
</tr>
</tbody>
</table>

**Query Syntax**  
CORR:DUT:SPOT:SPEC?

**Return Type**  
Character

**Default**  
FIX

---

Last Modified:

01June2018 Added NF mode Opt. 356 (10.3)
[:SENSe]:CORRection:ENR:EXTRapolated?

*(Read-Only)* Queries the ENR table's extrapolation status (Extrapolation Enabled (1)/ Extrapolation Disabled (0).

**Note:** The ENR extrapolation status is only relevant when the *CORRection:ENR:MODe* is set to Table.

See also, [:SENSe]:CORRection:UCALibration:CANCel, [:SENSe]:CORRection:UCALibration:RUN, and [:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:RCALibration:CANCel, [:SENSe]:CORRection:UCALibration[:STATe], [:SENSe]:CORRection:USER:INTerpolated?, and [:SENSe]:CORRection:RCALibration[:STATe]?

*Relevant Modes* **NF**

*Parameters* <not applicable>

*Examples* n/a

*Query Syntax* CORR:USER:EXTR?

*Return Type* 0 (False) - Extrapolation is disabled
1 (True) - Extrapolation is enabled

*Default* not applicable

---

[:SENSe]:CORRection:ENR:MODe <char>

*(Read-Write)* Set and query the noise source ENR mode setting.

See also [:SENSe]:CORRection:ENR:SPOT:ENR, [:SENSe]:CORRection:ENR:UNCertainty, [:SENSe]:CORRection:ENR:ONGamma, [:SENSe]:CORRection:ENR:OFFGamma, [:SENSe]:CORRection:ENR:COVerage, [:SENSe]:CORRection:ENR:SPECify, [:SENSe]:CORRection:ENR:DISTribution, and [:SENSe]:CORRection:TCOLd.

*Relevant Modes* **NF**

*Parameters* <character> ENR mode. Choose from:
- **SPOT** - Sets the ENR mode to Spot, where you can customize the ENR values for your device.
- **TABL** - Time per point can be set, but other integration SCPIs are not settable.

*Examples* CORR:ENR:MOD SPOT
CORR:ENR:MOD TABL

*Query Syntax* CORR:ENR:MODe?

*Return Type* Character
[:SENSe]:CORRection:ENR:SPOT:COVerage <char>

(Read-Write) Set and query the value of the ENR uncertainty coverage (1σ, 2σ, or 3σ).

Note: Only available when CORR:ENR:MOD is set to SPOT.


Relevant Modes NF

Parameters

<character> ENR Spot mode's ENR uncertainty coverage value. Choose from:

SD1 - 1σ (About 68% of the values fall within 1 standard deviation of the mean.)

SD2 - 2σ (About 95% of the values fall within 2 standard deviations of the mean.)

SD3 - 3σ (About 99.7% of the values fall within 3 standard deviations of the mean.)

Examples

CORR:ENR:SPOT:COV SD1
CORR:ENR:SPOT:COV SD3

Query Syntax CORR:ENR:SPOT:COV?

Return Type Character

Default SD2
Relevant Modes  
**NF**

Parameters

<character> ENR Spot mode's distribution value. Choose from:

- **RAYL** - The probability density function (pdf) of the magnitude of the reflection coefficient is a Rayleigh distribution. This is normally the case for specified values when the overall mismatch is due to several interacting impedance transitions within the device.

- **UNIF** - A distribution that was historically used in the industry, before the Rayleigh research, as documented in Keysight Application Note 1449-3, but is rarely accurate. Refer to [https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf](https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf).

- **FIX** - The reflection coefficient magnitude is always at the given level. This is often the case when measured values are used.

Examples

<table>
<thead>
<tr>
<th>CORR:ENR:DIST FIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:ENR:DIST RAYL</td>
</tr>
</tbody>
</table>

Query Syntax  
CORRection:ENR:DIST?

Return Type  
Character

Default  
FIX

Last Modified:

01june2018  
Added NF mode Opt. 356 (10.3)

>::SENSe]:CORRection:ENR:SPOT:ENR <num>

(Read-Write) Set and query the ENR Spot value in dB.

Note: Only available when CORR:ENR:MOD is set to SPOT.

See also  
>::SENSe]:CORRection:ENR:MODE, >::SENSe]:CORRection:ENR:SPOT:UNCertainty, >::SENSe]:CORRection:ENR:SPOT:ONGamma, >::SENSe]:CORRection:ENR:SPOT:OFFGamma, >::SENSe]:CORRection:ENR:SPOT:COVerage, >::SENSe]:CORRection:ENR:SPOT:SPECify, >::SENSe]:CORRection:ENR:SPOT:DISTRIBUTION, and >::SENSe]:CORRection:TCOLd.

Relevant Modes  
**NF**

Parameters

<num> ENR Spot mode. Choose from:

Minimum of -100 dB up to
Maximum of 100 dB

Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:ENR:SPOT:ENR</td>
<td>20</td>
</tr>
<tr>
<td>CORR:ENR:SPOT:ENR</td>
<td>-10.507</td>
</tr>
</tbody>
</table>

Query Syntax
CORR:ENR:SPOT:ENR?

Return Type
Numeric

Default
1.5000+01

Last Modified:
01 June 2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:ENR:SPOT:OFF:Gamma <num>

(Read-Write) Set and query the Off value (cold) of the ENR spot value’s 50Ω match, \( \Gamma \) (0.000 to 1.000).

Note: Only available when CORR:ENR:MOD is set to SPOT.


Relevant Modes
NF

Parameters

<numeric>
ENR Spot mode's 50Ω match, \( \Gamma \) Off value (cold). Choose from:
- Minimum of 0.000 up to
- Maximum of 1

Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:ENR:SPOT:OFFG</td>
<td>.2</td>
</tr>
<tr>
<td>CORR:ENR:SPOT:OFFG</td>
<td>0.347</td>
</tr>
</tbody>
</table>

Query Syntax
CORR:ENR:SPOT:OFFG?

Return Type
Numeric

Default
0.000E+00

Last Modified:
01 June 2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:ENR:SPOT:ONGamma <num>
(Read-Write) Set and query the On value (hot) of the ENR spot value’s 50Ω match, Γ (0.000 to 1.000).

Note: Only available when CORR:ENR:MOD is set to SPOT.


Relevant Modes  NF

Parameters

<numeric>  ENR Spot mode's 50Ω match, Γ On value (hot). Choose from:
Minimum of 0.000 up to
Maximum of 1

Examples  CORR:ENR:SPOT:ONG .2
CORR:ENR:SPOT:ONG 0.347

Query Syntax  CORRection:ENR:SPOT:ONGamma?

Return Type  Numeric

Default  0.000E+00

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:ENR:SPOT:SPEC <char>

(Read-Write) Set and query the ENR specified fixed value. This value indicates the nature of the entered value relative to the population of values it represents.

Note: Only available when CORR:ENR:MOD is set to SPOT.

See also [:SENSe]:CORRection:ENR:MODE, [:SENSe]:CORRection:ENR:SPOT:ENR, [:SENSe]:CORRection:ENR:SPOT:UNCertainty, [:SENSe]:CORRection:ENR:SPOT:OFFGamma, [:SENSe]:CORRection:ENR:SPOT:COVerage, [:SENSe]:CORRection:ENR:SPOT:DISTribution, and [:SENSe]:CORRection:TCOLd.

Relevant Modes  NF

Parameters

<character>  ENR Spot mode's fixed value. Choose from:

MAX  - Use this if this value represents the largest possible reflection coefficient for the population of devices to which the DUT belongs. This is true for "does not exceed" specifications.

PCTL95  - Use this if 95% of the population has a lower reflection coefficient than the value entered. This is the case for "2 standard deviation" (2s) type
specifications.

**PCTL80** - Use this if 80% of the population has a lower reflection coefficient than the value entered.

**MED** or **MEAN** - Use this if this value represents the median / mean reflection coefficient for the population of devices to which the DUT belongs.

**FIX** - Use this if the value is a measured value for the DUT.

**Examples**

```
CORR:ENR:SPOT:SPEC PCTL80
CORR:ENR:SPOT:SPEC MED
```

**Query Syntax**

```
CORR:ENR:SPOT:SPEC?
```

**Return Type** Character

**Default** FIX

---

Last Modified: 01june2018 Added NF mode Opt. 356 (10.3)

### [:SENSe]:CORRRection:ENR:SPOT:UNCertainty <num>

*(Read-Write)* Set and query the spot value ENR uncertainty value in dB.

Note: Only available when **CORR:ENR:MOD** is set to SPOT.


**Relevant Modes** NF

**Parameters**

- `<numeric>` ENR Spot mode's uncertainty value. Choose from:
  - Minimum of -100 dB up to
  - Maximum of 100 dB

**Examples**

```
CORR:ENR:SPOT:UNC 20
CORR:ENR:SPOT:UNC -10.507
```

**Query Syntax**

```
CORR:ENR:SPOT:UNCertainty?
```

**Return Type** Numeric

**Default** 0.000E+00

---

Last Modified: 01june2018 Added NF mode Opt. 356 (10.3)
(Read-Write) Set and query the port extension value on port 1.
Use [:SENSe]:CORRection:EXTension[:STATe] to turn port extensions ON and OFF.
Use [:SENSe]:CORRection:RVElocity:COAX to set velocity factor.

Relevant Modes  CAT, NA
Parameters

<bool>  Port extension in seconds. Choose a value between -10.0 to +10.0.
Examples  CORR:EXT:PORT1 1e-10
Query Syntax  [:SENSe]:CORRection:EXTension:PORT1?
Return Type  Numeric
Default  0

Last Modified:
10-Aug-2010  New command (5.30)

[:SENSe]:CORRection:EXTension:PORT2 <num>

(Read-Write) Set and query the port extension value on port 2.
Use [:SENSe]:CORRection:EXTension[:STATe] to turn port extensions ON and OFF.
Use [:SENSe]:CORRection:RVElocity:COAX to set velocity factor.

Relevant Modes  CAT, NA
Parameters

<bool>  Port extension in seconds. Choose a value between -10.0 to +10.0.
Examples  CORR:EXT:PORT2 1e-10
Query Syntax  [:SENSe]:CORRection:EXTension:PORT2?
Return Type  Numeric
Default  0

Last Modified:
10-Aug-2010  New command (5.30)

[:SENSe]:CORRection:EXTension[:STATe] <bool>

(Read-Write) Set and query the port extension ON | OFF state.
Use [:SENSe]:CORRection:EXTension:PORT1 and [:SENSe]:CORRection:EXTension:PORT2 to set port extensions value.
Use [:SENSe]:CORRection:RVELocity:COAX to set velocity factor.

**Relevant Modes**  CAT, NA

**Parameters**

- `<bool>`  Port extensions state. Choose from:
  - 0 or OFF - Port extensions OFF
  - 1 or ON - Port extensions ON

**Examples**  CORR:EXT  1

**Query Syntax**  [:SENSe]:CORRection:EXTension[:STATe]?

**Return Type**  Boolean

**Default**  OFF

---

Last Modified:

10-Aug-2010  New command (5.30)

---

[:SENSe]:CORRection:GAIN2[:INPut][:MAGNitude] <num>

*(Read-Write)* Set and query the Power Meter Offset value.

**Relevant Modes**  Power Meter, Pulse Measurements, CPM

**Parameters**

- `<bool>`  Offset value in dB. A positive value compensates for a component with loss. A negative value compensates for a component with gain.

**Examples**  CORR:GAIN2  3

**Query Syntax**  [:SENSe]:CORRection:GAIN2?

**Return Type**  Numeric

**Default**  0

---

Last modified:

1-Apr-2014  Added CPM

31-Oct-2013  Added Pulse

---

[:SENSe]:CORRection:GAIN2:STATe <bool>

*(Read-Write)* Set and query the ON | Off state of Power Meter Offset.

**Relevant Modes**  Power Meter, Pulse Measurements, CPM
Parameters

<bool> Offset state. Choose from:
  0 or OFF - Offset OFF
  1 or ON - Offset ON

Examples  CORR:GAIN2:STAT 1

Query Syntax  [:SENSe]:CORRection:GAIN2:STATe?

Return Type  Boolean

Default  0

Last modified:
  1-Apr-2014  Added CPM
  31-Oct-2013  Added Pulse

[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] <num>

(Read-Write) Set and query the system impedance.

Relevant Modes  NA

Parameters

<num> System impedance. Choose either 50 or 75.

Examples  CORR:IMP 75

Query Syntax  [:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?

Return Type  Numeric

Default  50

[:SENSe]:CORRection:LOSS:AFTer:ENABled <bool>

(Read-Write) Set and query the enable/disable state of the noise figure correction loss after the DUT.

Use CORRection:LOSS:AFTer to set the correction loss level after the DUT (DUT Out).

Relevant Modes  NF

Parameters

<bool> Enable/Disable the DUT Out state. Choose from:
  ON (or 1) - The correction loss value stated using the CORRection:LOSS:AFTer command.
  OFF (or 0) - The correction loss after the DUT is disabled.

Examples  :CORR:LOSS:AFT:ENAB 1
correction:loss:after:enabled off
**Query Syntax**  :CORRection:LOSS:AFTer:ENABled?
**Return Type**  Boolean
**Default**  OFF

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)

[:SENSe]:CORRection:LOSS:AFTer[:VALue] <double>

*(Read-Write)* Set and query the amount of noise figure correction loss after the DUT input.
Use **CORRection:LOSS:AFTer:ENABled** to set the correction loss level after the DUT input.

**Relevant Modes**  NF

**Parameters**
- **<double>**  Set the amount of noise figure loss correction after the DUT input. Choose from:
  - **-100 to 100 dB**  - The correction loss value stated using the **CORRection:LOSS:AFTer:ENABled** command.
  - **0.00 dB**  - Default

**Examples**  :CORR:LOSS:AFT -15.55

**Query Syntax**  :CORRection:LOSS:AFTer?
**Return Type**  Double
**Default**  OFF

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)

[:SENSe]:CORRection:LOSS:BEFore:ENABled <bool>

*(Read-Write)* Set and query the enable/disable state of the noise figure correction loss before the DUT (DUT In).
Use **CORRection:LOSS:BEFore[:VALue]** to set the correction loss level before the DUT (DUT In).

**Relevant Modes**  NF

**Parameters**
- **<bool>**  Enable/Disable the DUT In state. Choose from:
  - **ON (or 1)**  - The correction loss value stated using the **CORRection:LOSS:BEFore[:VALue]** command.
**OFF (or 0)** - The correction loss before the DUT (DUT In) is disabled.

**Examples**

```plaintext
:CORR:LOSS:BEF:ENABL 1
  off
```

**Query Syntax**

`:CORR:LOSS:BEF:ENABled?`

**Return Type**

Boolean

**Default**

OFF

---

Last Modified:

01-june-2018  Added Opt. 356 NF(10.3)

**[:SENSe]:CORRection:LOSS:BEFore[:VALue] <double>**

*(Read-Write)* Set and query the amount of noise figure correction loss before the DUT input. Use **CORRection:LOSS:BEfore:ENABled** to set the correction loss level before the DUT input.

**Relevant Modes**

NF

**Parameters**

`<double>` - Set the amount of noise figure loss correction before the DUT input. Choose from:

- **-100 to 100 dB** - The correction loss value stated using the **CORRection:LOSS:BEFORE:ENABled** command.
- **0.00 dB** - Default

**Examples**

```plaintext
:CORR:LOSS:BEF -15.55
```

**Query Syntax**

`:CORR:LOSS:BEF?`

**Return Type**

Double

**Default**

OFF

---

Last Modified:

01-june-2018  Added Opt. 356 NF(10.3)

**[:SENSe]:CORRection:LOSS:COAX <num>**

*(Read-Write)* Set and query the electrical loss of transmission cable to be used with DTF measurements.

**Relevant Modes**

CAT

**Parameters**
Loss in dB.

**Examples**

CORR:LOSS:COAX .5

**Query Syntax**

[:SENSe]:CORRection:LOSS:COAX?

**Return Type**

Numeric

**Default**

0

[:SENSe]:CORRection:LOSS:WAVeguide

Type topic text here.

[:SENSe]:CORRection:MEDium <char>

(Write-Read) Set and query the calibration media type.

If the measurement requires electrical delay or port extensions, available ONLY in NA Mode, you must specify the waveguide Media Type and Cutoff Frequency. These settings are needed to help specify dispersion. They are NOT used if electrical delay and port extensions are both zero.

**Relevant Modes**

NA, CAT

**Parameters**

- **<bool>** Media type. Choose from the following:
  - COAX - Calibration standards are coaxial.
  - WAVeguide - Calibration standards are waveguide.

**Examples**

CORR:MED WAV

Sense:Correction:Medium Waveguide

See Cal examples

**Query Syntax**

[:SENSe]:CORRection:MEDium?

**Default**

COAX

Last Modified:

1-Nov-2013 New command

[:SENSe]:CORRection:PAMPLifier:MODe <char>

(Read-Write) Set and query the Preamplifier uncertainty setup values being measured.

See also [:SENSe]:CORRection:PAMPLifier:INGamma, [:SENSe]:CORRection:PAMPLifier:OUTGamma, [:SENSe]:CORRection:PAMPLifier:SPEC, [:SENSe]:CORRection:PAMPLifier:DISTribution.

See also, MMEMory:LOAD:PAMPLifier and MMEMory:STORe:PAMPLifier.

**Relevant Modes**

NF

**Parameters**
Preamplifier mode uncertainty settings. Choose from:

**SPOT** - Sets the Preamplifier mode to Spot, where you can customize the ENR values for your device.

**TABL** - Time per point can be set, but other integration SCPIs are not settable.

### Examples

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:PAMPL:MOD SPOT</td>
</tr>
<tr>
<td>CORR:PAMPL:MOD TABL</td>
</tr>
</tbody>
</table>

### Query Syntax

CORRection:PAMPL:MODe?

### Return Type

Character

### Default

SPOT

---

**[:SENSe]:CORRection:PAMPLifier:SPOT:DISTribution** <char>

(Read-Write) Set and query the preamplifier specified fixed value. This value indicates the nature of the entered value relative to the population of values it represents. (i.e., Fixed, Rayleigh, or Uniform (Uniform in Circle).

Note: Only available when CORR:PAMP:MOD is set to SPOT.

See also [:SENSe]:CORRection:PAMPLifier:SPOT:INGamma, [:SENSe]:CORRection:PAMPLifier:SPOT:OUTGamma, [:SENSe]:CORRection:PAMPLifier:MODE, [:SENSe]:CORRection:PAMPLifier:SPOT:SPEC.

See also, MMEMory:LOAD:PAMPLifier and MMEMory:STORe:PAMPLifier.

### Relevant Modes

NF

### Parameters

**<character>** Preamplifier Spot mode's distribution value. Choose from:

**RAYL** - The probability density function (pdf) of the magnitude of the reflection coefficient is a Rayleigh distribution. This is normally the case for specified values when the overall mismatch is due to several interacting impedance transitions within the device.

**UNIF** - A distribution that was historically used in the industry, before the Rayleigh research, as documented in Keysight Application Note 1449-3, but is rarely accurate. Refer to [https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf](https://literature.cdn.keysight.com/litweb/pdf/5988-9215EN.pdf).

**FIX** - The reflection coefficient magnitude is always at the given level. This is often the case when measured values are used.

### Examples

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:PAMP:SPOT:DIST FIX</td>
</tr>
<tr>
<td>CORR:PAMP:SPOT:DIST RAYL</td>
</tr>
</tbody>
</table>

### Query Syntax

CORRection:PAMP:SPOT:DIST?
[:SENSe]:CORRection:PAMPLifier:SPOT:INGamma <num>

(Read-Write) Set and query the preamplifier's input 50Ω match, Γ (0.000 to 1.000).
Note: Only available when CORR:PAMP:MOD is set to SPOT.
See also [:SENSe]:CORRection:PAMPLifier:MODE, [:SENSe]:CORRection:PAMPLifier:SPOT:OUTGamma, [:SENSe]:CORRection:PAMPLifier:SPOT:SPEC, [:SENSe]:CORRection:PAMPLifier:SPOT:DISTRibution.
See also, MMEMory:LOAD:PAMPLifier and MMEMory:STORe:PAMPLifier.

Relevant Modes NF

Parameters
<numeric> Preamplifier spot mode's input 50Ω match, Γ value. Choose from:
Minimum of 0.000 up to
Maximum of 1

Examples CORR:PAMP:SPOT:ING .2
CORR:PAMP:SPOT:ING 0.347

Query Syntax CORR:PA M:SPOT:INGamma?

Return Type Numeric
Default 0.000E+00

Last Modified: 01june2018 Added NF mode Opt. 356 (10.3)
(Read-Write) Set and query the preamplifier's output 50Ω match, $\Gamma$ (0.000 to 1.000).

Note: Only available when CORR:PAMP:MODE is set to SPOT.

See also [:SENSe]:CORRection:PA MPLifier:SPOT:INGamma, [:SENSe]:CORRection:PA MPLifier:MODE, [:SENSe]:CORRection:PA MPLifier:SPOT:SPEC, [:SENSe]:CORRection:PA MPLifier:SPOT:DISTRibution.

See also, MMEMory:LOAD:PA MPLifier and MMEMory:STORe:PA MPLifier.

Relevant Modes  **NF**

Parameters

<numeric>  Preamplifier spot mode's, output 50Ω match, $\Gamma$ value. Choose from:
Minimum of 0.000 up to
Maximum of 1

Examples  
CORR:PAMP:SPOT:OUTG .2
CORR:PAMP:SPOT:OUTG 0.347

Query Syntax  CORRection:PAMP:SPOT:OUTGamma?

Return Type  Numeric

Default  0.000E+00

Last Modified:
01june2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:PA MPLifier:SPOT:SPEC <char>

(Read-Write) Set and query the preamplifier's specified fixed value. This value indicates the nature of the entered value relative to the population of values it represents.

Note: Only available when CORR:PAMP:MODE is set to SPOT.

See also [:SENSe]:CORRection:AMPLifier:SPOT:INGamma, [:SENSe]:CORRection:PA MPLifier:SPOT:OUTGamma, [:SENSe]:CORRection:PA MPLifier:MODE, [:SENSe]:CORRection:PA MPLifier:SPOT:DISTRibution.

See also, MMEMory:LOAD:PA MPLifier and MMEMory:STORe:PA MPLifier.

Relevant Modes  **NF**

Parameters

<character>  Preamplifier Spot mode's fixed value. Choose from:

- **MAX** - Use this if this value represents the largest possible reflection coefficient for the population of devices to which the preamplifier belongs. This is true for "does not exceed" specifications.

- **PCTL95** - Use this if 95% of the population has a lower reflection coefficient than the value entered. This is the case for "2 standard deviation" (2s) type specifications.

- **PCTL80** - Use this if 80% of the population has a lower reflection coefficient than the value entered.
**MED** or **MEAN** - Use this if this value represents the median / mean reflection coefficient for the population of devices to which the preamplifier belongs.

**FIX** - Use this if the value is a measured value for the preamplifier.

<table>
<thead>
<tr>
<th>Examples</th>
<th>CORR:AMP:SPOT:SPEC PCTL95</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CORR:AMP:SPOT:SPEC MED</td>
</tr>
</tbody>
</table>

**Query Syntax**

CORRrection:AMP:SPOT:SPEC?

**Return Type**

Character

**Default**

FIX

---

**[:SENSe]:CORRection:RCALibration:CANCe!**

*(Write-Only) Cancels the receiver calibration.*

See also, [:SENSe]:CORRection:UCALibration:CANCe!, [:SENSe]:CORRection:UCALibration:RUN,

[:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:UCALibration[:STATe],

[:SENSe]:CORRection:UCALibration:INTerpolated?, [:SENSe]:CORRection:ENR:EXTRapolated?, and

[:SENSe]:CORRection:RCALibration[:STATe]?

**Note:** You can use the IEEE command *OPC? to query the status of your calibrations.

**Relevant Modes**

NF

**Parameters**

*not applicable*

<table>
<thead>
<tr>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:RCAL:CANC</td>
</tr>
</tbody>
</table>

*Cancels the receiver calibration, when a SCPI is used to run the Receiver calibration.*

**Query Syntax**

no query

**Return Type**

not applicable

**Default**

not applicable

---

**[:SENSe]:CORRection:RCALibration:RUN**

*(Write-Only) Runs the receiver calibration.*

See also, [:SENSe]:CORRection:UCALibration:CANCe!, [:SENSe]:CORRection:UCALibration:RUN,

[:SENSe]:CORRection:RCALibration:CANCe!, [:SENSe]:CORRection:UCALibration[:STATe],

[:SENSe]:CORRection:ENR:EXTRapolated?, [:SENSe]:CORRection:UCALibration:INTerpolated?, and

---

01June2018 Added NF mode Opt. 356 (10.3)
### [:SENSe]:CORRection:RCAlibration[:STATe]? 

Note: You can use the IEEE command *OPC? to query the status of your calibrations.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;not applicable&gt;</td>
</tr>
</tbody>
</table>

**Examples**

```
CORR:RCAL:RUN 'Runs the user calibration'
```

**Query Syntax**

`CORR:RCAL?`

**Return Type**

- `0 (False)` - Receiver calibration is disabled
- `1 (True)` - Receiver calibration is enabled

**Default**

`not applicable`

---

Last Modified:
01-june-2018 Added Opt. 356 NF(10.3)

---

### [:SENSe]:CORRection:RVELocity:COAX <num>

- **(Read-Only)** Queries the receiver calibration's status (Enabled (1)/Disabled (0)).

See also, [:SENSe]:CORRection:UCAlibration:CANCEL, [:SENSe]:CORRection:UCAlibration:RUN, [:SENSe]:CORRection:RCAlibration:RUN, [:SENSe]:CORRection:RCAlibration:CANCEL, [:SENSe]:CORRection:UCAlibration:INTERpolated?, and [:SENSe]:CORRection:ENR:EXTRapolated?.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>NF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;not applicable&gt;</td>
</tr>
</tbody>
</table>

**Examples**

`n/a`

**Query Syntax**

`CORR:RCAL?`

**Return Type**

- `0 (False)` - Receiver calibration is disabled
- `1 (True)` - Receiver calibration is enabled

**Default**

`not applicable`

---

Last Modified:
01-june-2018 Added Opt. 356 NF(10.3)
(Read-Write) Set and query the velocity factor to be used with DTF measurements (CAT mode) and with Port Extensions (NA mode).

**Relevant Modes**  
CAT, NA

**Parameters**

<num>  
Velocity factor. Choose a number between 0 and 1  
.66 = polyethylene dielectric  
.7 = PTFE dielectric

**Examples**

CORR : RVEL : COAX .7

**Query Syntax**

[:SENSe]: CORRection:RVELocity:COAX?

**Return Type**  
Numeric

**Default**  
1

Last Modified:

27-Apr-2012  
Edited for PTFE

10-Aug-2010  
New command for NA mode (5.30)

[SENSe]:CORRection[:STATe] <bool>

(Read-Write) Set and query the correction ON | OFF state.  
This will turn ALL calibration OFF, including "Cal Ready". This can NOT be done from the user-interface.  
See also [:SENSe]:CORRection:USER[:STATe]

**Relevant Modes**  
CAT, NA

**Parameters**

<bool>  
Correction state. Choose from:  
0 or OFF - Error Correction OFF  
1 or ON - Error Correction ON

**Examples**

CORR 1

**Query Syntax**

[:SENSe]: CORRection[:STATe]?

**Return Type**  
Boolean

**Default**  
ON

Last Modified:

18-Oct-2012  
Edited for new models
(Read-Write) Set and query the ENR temperature (cold). All temperature values Centigrade and Fahrenheit are converted to Kelvin.

See also [:SENSe]:CORRection:ENR:MODe, [:SENSe]:CORRection:ENR:SPOT:ENR, [:SENSe]:CORRection:ENR:UNCertainty, [:SENSe]:CORRection:ENR:ONGamma, [:SENSe]:CORRection:ENR:COVerage, [:SENSe]:CORRection:ENR:SPECify, [:SENSe]:CORRection:ENR:DISTribution, and [:SENSe]:CORRection:ENR:OFFGamma.

Relevant Modes  
NF
Parameters

<numeric>  ENR temperature (cold). Choose from:
Minimum of 0.000 up to
Maximum of 2.965 MK

Examples  
CORR:TCOL .2
CORR:TCOL 0.347

Query Syntax  CORRection:TCOLd?
Return Type  Numeric
Default  2.9650E+02

Last Modified:
01-june-2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:UCALibration:CANCel

(Write-Only) Cancels the user calibration.

See also, [:SENSe]:CORRection:UCALibration:RUN, [:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:RCALibration:CANCel, [:SENSe]:CORRection:UCALibration:INTerpolated?, [:SENSe]:CORRection:ENR:EXTRapolated?, [:SENSe]:CORRection:UCALibration[:STATe], and [:SENSe]:CORRection:RCALibration[:STATe]?

Note: You can use the IEEE command *:OPC? to query the status of your calibrations.

Relevant Modes  NF
Parameters

<not applicable>

Examples  CORR:UCAL:CALC 'Cancels the user calibration, when a SCPI is used to run the User calibration.'

Query Syntax  no query
Return Type  not applicable
Default  not applicable

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)
[:SENSe]:CORRection:UCALibration:INTerpolated?

(Read-Only) Queries the user calibration's interpolation status (Interpolation Enabled (1)/ Interpolation Disabled (0).

Note: The User Cal interpolation is only relevant when a User Cal is applied.

See also, [:SENSe]:CORRection:UCALibration:CANCel, [:SENSe]:CORRection:UCALibration:RUN, [:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:RCALibration:CANCel, [:SENSe]:CORRection:UCALibration[:STATe], [:SENSe]:CORRection:ENR:EXTRapolated?, and [:SENSe]:CORRection:RCALibration[:STATe]?

Relevant Modes  NF
Parameters  <not applicable>
Examples  n/a
Query Syntax  CORR:UCAL:INT?
Return Type  0 (False) - Interpolation is disabled
1 (True) - Interpolation is enabled
Default  not applicable

Last Modified: 01-june-2018  Added Opt. 356 NF(10.3)

[:SENSe]:CORRection:UCALibration:RUN

(Write-Only) Runs the user calibration.

See also, [:SENSe]:CORRection:UCALibration:CANCel, [:SENSe]:CORRection:UCALibration[:STATe], [:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:RCALibration:CANCel, [:SENSe]:CORRection:UCALibration:INTerpolated?, [:SENSe]:CORRection:ENR:EXTRapolated?, and [:SENSe]:CORRection:RCALibration[:STATe]?

Note: You can use the IEEE command *OPC? to query the status of your calibrations.

Relevant Modes  NF
Parameters  <not applicable>
Examples  CORR:UCAL:RUN  'Runs the user calibration
Query Syntax  no query
Return Type  not applicable
[:SENSe]:CORRection:UCALibration[:STAtE] <bool>

(Read-Write) Set and query the Apply User Calibration (correction) ON | OFF state.

See also, [:SENSe]:CORRection:UCALibration:CANCEL, [:SENSe]:CORRection:UCALibration:RUN, and [:SENSe]:CORRection:RCALibration:RUN, [:SENSe]:CORRection:RCALibration:CANCEL, [:SENSe]:CORRection:UCALibration:INTERpolated?, [:SENSe]:CORRection:ENR:EXTRapolated?, and [:SENSe]:CORRection:RCALibration[:STAtE]?

Relevant Modes NF

Parameters

<bool> User calibration state. Choose from:
0 or OFF - User calibration OFF (do not apply)
1 or ON - User calibration ON (do not apply)

Examples CORR:UCAL 1

Query Syntax [:SENSe]:CORRection:UCALibration[:STAtE]?

Return Type Boolean

Default On after calibration

Last Modified:
01-june-2018 Added Opt. 356 NF(10.3) and new command

[:SENSe]:CORRection:UNCertainty:BARS

(Read-Write) Set and query the uncertainty bars ON | OFF state.

See also, [:SENSe]:CORRection:UNCertainty:JITTER, [:SENSe]:CORRection:UNCertainty:MIStMatch, [:SENSe]:CORRection:UNCertainty:ENR, [:SENSe]:CORRection:UNCertainty:COVerage, and [:SENSe]:CORRection:UNCertainty:CALibration

For trace-related uncertainty commands, refer to CALCulate:[SELected]TRACe:UNCertainty:DATA?, CALCulate:[SELected]:TRACe:UNCertainty:UPPer:DATA?, and CALCulate:[SELected]:TRACe:UNCertainty:LOWer:DATA?.

Relevant Modes NF

Parameters

<bool> Uncertainty bars state. Choose from:
396

0 or OFF - Uncertainty bars OFF
1 or ON - Uncertainty bars ON

Examples

CORR:UNC:BARS 1

Query Syntax

[:SENSe]:CORRection:UNCertainty:BARS?

Return Type

Boolean

Default

Off (0)

Last Modified:

18-Oct-2012  Edited for new models

[:SENSe]:CORRection:UNCertainty:CALibration <bool>

(Read-Write) Set and query the user calibration uncertainty ON | OFF state.

See also, [:SENSe]:CORRection:UNCertainty:JITTer, [:SENSe]:CORRection:UNCertainty:MISMatch,
[:SENSe]:CORRection:UNCertainty:ENR, [:SENSe]:CORRection:UNCertainty:COVerage, and
[:SENSe]:CORRection:UNCertainty:BARS.

For trace-related uncertainty commands, refer to CALCulate:[SE]El[ected]:TRACe:UNCertainty:DATA?,
CALCulate:[SE]El[ected]:TRACe:UNCertainty:UPPer:DATA?, and
CALCulate:[SE]El[ected]:TRACe:UNCertainty:LOWer:DATA?.

Relevant Modes    NF

Parameters

<bool> User calibration uncertainty state. Choose from:
0 or OFF - User calibration uncertainty OFF
1 or ON - User calibration uncertainty ON

Examples

CORR:UNC:CALC ON 'Can only be enabled when a user calibration has been completed'

Query Syntax

[:SENSe]:CORRection:UNCertainty:CALibration?

Return Type

Boolean

Default

Off (0)

Last Modified:

18-Oct-2012  Edited for new models

[:SENSe]:CORRection:UNCertainty:COVerage <char>
(Read-Write) Set and query the value of the current settings uncertainty contributions coverage (1σ, 2σ, or 3σ).

See also, [:SENSe]:CORRection:UNCertainty:JITter, [:SENSe]:CORRection:UNCertainty:MISMatch, [:SENSe]:CORRection:UNCertainty:ENR, [:SENSe]:CORRection:UNCertainty:CALibration, and [:SENSe]:CORRection:UNCertainty:BARS

For trace-related uncertainty commands, refer to CALCulate:[SELelected]:TRACe:UNCertainty:DATA?, CALCulate:[SELelected]:TRACe:UNCertainty:UPPer:DATA?, and CALCulate:[SELelected]:TRACe:UNCertainty:LOWer:DATA?.

**Relevant Modes** NF

**Parameters**

<character> Current settings uncertainty coverage value. Choose from:

- **SD1** - 1σ (About 68% of the values fall within 1 standard deviation of the mean.)
- **SD2** - 2σ (About 95% of the values fall within 2 standard deviations of the mean.)
- **SD3** - 3σ (About 99.7% of the values fall within 3 standard deviations of the mean.)

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR:UNC:COV SD1</td>
<td></td>
</tr>
<tr>
<td>CORR:UNC:COV SD3</td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax** CORRection:UNC:COV?

**Return Type** Character

**Default** SD2

---

Last Modified:

01june2018 Added NF mode Opt. 356 (10.3)

---

[:SENSe]:CORRection:UNCertainty:ENR<bool>

(Read-Write) Set and query the ENR source uncertainty ON | OFF state.

See also, [:SENSe]:CORRection:UNCertainty:JITter, [:SENSe]:CORRection:UNCertainty:MISMatch, [:SENSe]:CORRection:UNCertainty:CALibration, [:SENSe]:CORRection:UNCertainty:COVerage, and [:SENSe]:CORRection:UNCertainty:BARS

For trace-related uncertainty commands, refer to CALCulate:[SELelected]:TRACe:UNCertainty:DATA?, CALCulate:[SELelected]:TRACe:UNCertainty:UPPer:DATA?, and CALCulate:[SELelected]:TRACe:UNCertainty:LOWer:DATA?.

**Relevant Modes** NF

**Parameters**

<bool> ENR source uncertainty state. Choose from:

- **0 or OFF** - Source uncertainty OFF
Examples

CORR:UNC:ENR 0

Query Syntax
[:SENSe]:CORRection:UNCertainty:ENR?

Return Type
Boolean

Default
On (1)

Last Modified:
18-Oct-2012 Edited for new models

[:SENSe]:CORRection:UNCertainty:JITTer <bool>

(Read-Write) Set and query the jitter uncertainty ON | OFF state.

See also, [:SENSe]:CORRection:UNCertainty:ENR, [:SENSe]:CORRection:UNCertainty:MISMatch, [:SENSe]:CORRection:UNCertainty:CALibration, [:SENSe]:CORRection:UNCertainty:COVerage, and [:SENSe]:CORRection:UNCertainty:BARS


Relevant Modes
NF

Parameters

<bool> Jitter uncertainty state. Choose from:
0 or OFF - Jitter OFF
1 or ON - Jitter uncertainty ON

Examples
CORR:UNC:JITT 0

Query Syntax
[:SENSe]:CORRection:UNCertainty:JITTer?

Return Type
Boolean

Default
On (1)

Last Modified:
18-Oct-2012 Edited for new models

[:SENSe]:CORRection:UNCertainty:MISMatch <bool>

(Read-Write) Set and query the mismatch uncertainty ON | OFF state.

See also, [:SENSe]:CORRection:UNCertainty:JITTer, [:SENSe]:CORRection:UNCertainty:ENR, [:SENSe]:CORRection:UNCertainty:CALibration, [:SENSe]:CORRection:UNCertainty:COVerage, and [:SENSe]:CORRection:UNCertainty:BARS

For trace-related uncertainty commands, refer to CALCulate:[SELeeted]TRACe:UNCertainty:DATA?, CALCulate:[SELeeted]:TRACe:UNCertainty:UPPer:DATA?, and
CALCulate:[SELection]:TRACe:UNCertainty:LOWer:DATA?

Relevant Modes  NF

Parameters
<bool>  Mismatch uncertainty state. Choose from:
  0 or OFF - Mismatch uncertainty OFF
  1 or ON - Mismatch uncertainty ON

Examples  CORR:UNC:MISM 0

Query Syntax  [:SENSe]:CORRection:UNCertainty:MISMatch?

Return Type  Boolean

Default  On (1)

Last Modified:
18-Oct-2012  Edited for new models

[:SENSe]:CORRection:UNCertainty:QUEStionable?

(Read-Only)  Reads the status of the uncertainty questionable state (1 or 0).

Relevant Modes  NF

Examples  CORR:UNC:QUES?

Return Type  Numeric:
  0 (False) - The sweep uncertainty values are valid (not questionable)
  1 (True) - The sweep uncertainty values are questionable

Default  Not Applicable

Last Modified:
01June2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:CORRection:WAVeguide:STANdard <char>

(Read-Write)  Set and query the waveguide standard to use for the measurement. See also [:SENSe]:CORRection:MEDium.

Relevant Modes  CAT

Parameters
<char>  Waveguide standard. Choose from:
USER - A predefined user standard.
WR137
WR90
WR42
WR112

Examples
CORR:WAV:STAN WR137

Query Syntax
[:SENSe]:CORRection:WAVeguide:STANdard?

Return Type
Character

Default
FieldFox model dependent.

Last Modified:
25-Mar-2014 New command

[:SENSe]:CORRection:USER[:STATe] <bool>

(Read-Write) Set and query the User Calibration (correction) ON | OFF state.
N9912A - This includes Preset Cal correction.
All other models - This does NOT include "Cal Ready" correction, which can ONLY be turned OFF using [:SENSe]:CORRection[:STATe].

Relevant Modes
CAT, NA

Parameters
<bool> User calibration state. Choose from:
0 or OFF - User calibration OFF
1 or ON - User calibration ON

Examples
CORR 1

Query Syntax
[:SENSe]:CORRection:USER[:STATe]?

Return Type
Boolean

Default
On after calibration

Last Modified:
18-Oct-2012 Edited for new models

[:SENSe]:CORRection:WGCutoff <num>

(Read-Write) Set and query the waveguide cutoff frequency. This is the minimum frequency of the waveguide. See also [:SENSe]:CORRection:MEDium

Relevant Modes
NA, CAT
Parameters

<num> Waveguide cutoff frequency in Hz. The value is usually on the data sheet of the Waveguide Cal Kit.

Examples

CORR:WGC 100e6

Query Syntax [:SENSe]:CORRection:WGCutoff?

Return Type Numeric

Default 18 GHz

Last Modified:

1-Nov-2013 New command

[:SENSe]:CORRection:ZERO:REFerence <char>

(Read-Write) Set and query the reference to use when zero is performed.

Relevant Modes VVM

Parameters

<char> VVM Zero reference. Choose from:

OPEN - Display Phase = 0.0 Deg when zeroed.
SHORT - Display Phase = 180.0 Deg when zeroed

Examples

CORR: ZERO : REF SHOR

Query Syntax [:SENSe]:CORRection:ZERO:REFerence?

Return Type Character

Default OPEN

Last Modified:

22-Sept-2014 New command

[:SENSe]:CORRection:ZERO:STATe <char>

(Read-Write) Set and query the Zero State of a VVM measurement.

Relevant Modes VVM

Parameters

<char> VVM Zero state. Choose from:

OFF - Set Zeroing OFF
ON - Zero the measurement

Examples

CORR:ZERO:STAT OFF
Query Syntax: [:SENSe]:CORRection:ZERO:STATe?
Return Type: Character
Default: OFF

[:SENSe]:DIF:BANDwidth <num>

(Read-Write) Set and query the digital IF bandwidth of the instrument. This parameter is closely related to
the sample rate.

Relevant Modes: IQA

Parameters

<numeric> Minimum: 1
Maximum: 10000000

Examples:
DIF:BAND 1000
DIF:BAND 5 kHz
DIF:BAND 5e3

Query Syntax: :DIF:BAND?

Return Type: numeric
Default: 10000000

Last Modified: 22oct2017 New IQA command (A.10.1x)

[:SENSe]:DENSity:BPLLevel

(Read-Write) Set and query the Blue Limit percent of the RTSA Mode Density trace display state.

Relevant Modes: RTSA

Parameters

<num> Set the Blue limit percentage to a value between 0.05 and 100.
If you attempt to set the limit below 0.05, the instrument's blue limit will be set
to 0.05 percent.

Examples:
DENS:BPL 10 'Sets the blue limit percent to 10 percent.
DENS:BPL .05 'Sets the blue limit to its minimum (default) percent of 0.05.

Query Syntax: DENS:BPL?
Return Type: num
Default  5.0E-02

[:SENSe]:DENSity:RPLevel

(Read-Write) Set and query the red Limit percent of the RTSA Mode Density trace display state.

Relevant Modes  RTSA

Parameters

<num>  Set the red limit percentage to a value between 0.05 and 100.
If you attempt to set the limit below 0.05, the instrument's red limit will be set to 0.05 percent.

Examples

DENS:RPL 10  'Sets the red limit percent to 10 percent.
DENS:RPL .05  'Sets the red limit to its minimum (default) percent of 0.05.

Query Syntax  DENS RPL?

Return Type  num

Default  1.00E+02

Last Modified:

19-sep-2016  Added new cmd for RTSA mode (A.09.50).

[:SENSe]:DETector:FUNCtion <char>

(Read-Write) Set and query the SA detector function.

Relevant Modes  SA

Parameters

<bool>  Choose from:
AUTO
NORMal
POSitive
NEGative
SAMPle
AVERage

Examples

DET: FUNC POS

Query Syntax

[:SENSe]:DETector:FUNCtion?

Return Type

Character

Default

AUTO

[:SENSe]:DETector:TRACe{1:4}:FUNCtion

(Read-Write) Set and query the RTSA detector and trace (1:4) function.

Relevant Modes

RTSA

Parameters

<bool>
Choose from:

AUTO
NORMal
POSitive
NEGative
SAMPle
AVERage - (RMS Average)

Examples

DET:TRAC2:FUNC POS

Query Syntax

[:SENSe]:DETector:TRACe:FUNCtion?

Return Type

Character

Default

AUTO

Last Modified:

19-sep-2016 Added new command for RTSA (A.09.50).

[SENSe]:DUT:SIDEband <char>

(Read-Write) Set and query the sideband of the displayed trace.

See DUT setting commands for NF mode.

Relevant Modes

NF

Parameters

<char>
Choose from:

LSB - lower sideband
USB - upper sideband
DSB - double sideband
Examples
- DUT:SIDE LSB
- DUT:SIDE DSB

Query Syntax
SENSe:DUT:SIDEband?

Return Type
Character

Default
DUT  Type dependent. See :MODE:DUT.

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)

[:SENSe]:DUT[:TYPe] <char>

(Read-Write) Set and query the DUT mode (i.e., Amplifier, Downconverter, Upconverter, or Multi-Stage Converter).

See DUT setting commands for NF mode.

Relevant Modes
- NF

Parameters
- <char>

Choose from:
- AMPL - Amplifier
- DCON - Downconverter
- UCON - Upconverter
- MCON - Multi-Stage Converter

Examples
DUT AMP
DUT MCON

Query Syntax
SENSe:DUT:TYPe?

Return Type
Character

Default
AMPL

Last Modified:
01-june-2018  Added Opt. 356 NF(10.3)

[:SENSe]:FREQuency <num>

(Read-Write) Set and query the frequency of the power meter measurement. This is used to set the proper correction value of the power sensor.

Relevant Modes
- Power Meter, Pulse Measurements, CPM
Parameters

<num> Frequency in Hz. Choose a number between the minimum and maximum frequency of the FieldFox. This command will accept MIN and MAX as arguments.

Examples

FREQ 10E6
FREQ MAX

Query Syntax [:SENSe]:FREQuency?

Return Type Numeric

Default Center frequency of the FieldFox

Last Modified:
1-Apr-2014 Added CPM
29-Oct-2013 Added Pulse

[:SENSe]:FREQuency:ANNotation[:SELect] <char>

(Read-Write) Set and query the frequency (X-axis) annotation method.

See frequency setting commands for SA mode.

Relevant Modes SA, ERTA, RTSA

Parameters

<char> Choose from:
AUTO - Annotated the same as the frequencies were set.
SSTop - Start and Stop
CSPan - Center and Span

Examples

FREQ:ANN SST

Query Syntax [:SENSe]:FREQuency:ANNotation[:SELect]?

Return Type Character

Default AUTO

Last Modified:
10-June-2016 Added RTSA mode(9.50)
20-Jan-2015 New command (8.00)
[:SENSe]:FREQuency:CENTer <num>

(Read-Write) Set and query the center frequency of the trace. This is also used for single frequency (CW) traces, such as the frequency of power meter measurements.

**Parameters**
- `<numeric>`: Center frequency in Hz. Choose a number between the minimum and maximum frequency of the FieldFox.
  This command will accept MIN and MAX as arguments.

**Examples**
- `FREQ:CENT 10E6`
- `FREQ:CENT MAX`

**Query Syntax**
- `[:SENSe]:FREQuency:CENTer?`

**Return Type**
- Numeric

**Default**
- CAT, IQA, NA, NF, SA modes - Center frequency of the FieldFox.
- VVM mode - 2 MHz

---

Last Modified:
- 01-june-2018: Added Opt. 356 NF(10.3)
- 22-oct-2017: Added IQA mode (10.1x)
- 10-june-2016: Added RTSA mode (9.50)

[:SENSe]:FREQuency:CENTer:STEP <num>

(Read-Write) Set and query the step size for use with the front panel up/down keys.

Use `[:SENSe]:FREQuency:CENTer:STEP:AUTO` to set step size manually. (Not available with CPM).

**Parameters**
- `<num>`: Step size in Hz. Choose a number between 1 and maximum frequency of the FieldFox.

**Examples**
- `FREQ:CENT:STEP 10E6`

**Query Syntax**
- `[:SENSe]:FREQuency:CENTer:STEP?`

**Return Type**
- Numeric

**Default**
- Dependent on frequency span

---

Last Modified:
- 10-june-2016: Added RTSA mode (9.50)
- 1-Apr-2014: Added CPM
[:SENSe]:FREQuency:CENTer:STEP:AUTO <bool>

(Read-Write) Set and query the state of setting center frequency step size. Used with the front panel up/down keys.

**Relevant Modes**  SA, RTSA

**Parameters**

<bool>  Step size state. Choose from:

- **ON** or **1** - Step size is set automatically. Each press of the ▲|▼ arrows increments or decrements the value by 1/10th (one division) of the current frequency span.
- **OFF** or **0** - Step size is set manually with [:SENSe]:FREQuency:CENTer:STEP.

**Examples**

```
FREQ:CENT:STEP:AUTO 1
sense:frequency:center:step:auto off
```

**Query Syntax**  [:SENSe]:FREQuency:CENTer:STEP:AUTO?

**Return Type**  Boolean

**Default**  ON or 1

---

Last Modified:

10-june-2016  Added RTSA mode (9.50)

---

[SENSe]:FREQuency:CONText

(Read-Write) Set and query the current frequency context (RF or IF).

See DUT setting commands for NF mode.

**Relevant Modes**  NF

**Parameters**

<char>  Choose from:

- **RF** - Default when DUT Type = AMPL
- **IF** - Default when DUT Type = DOWN or UPC or MOCN

**Examples**

```
:FREQ:CONT RF
:FREQ:CONT IF
```

**Query Syntax**  :FREQuency:CONText?

**Return Type**  Character

**Default**  DUT Type dependent.

- **RF** - Default when DUT Type = AMPL
- **IF** - Default when DUT Type = DOWN or UPC or MOCN
[:SENSe]:FREQuency:DATA?

(Read-Write) Returns a comma separated array of the current x-axis frequency values. The size of the array equals the number of data points.

**Relevant Modes** CAT, NA, and NF

**Parameters** None

**Examples** FREQ:DATA?

**Return Type** Comma-separated numeric array

**Default** Not Applicable

Last Modified:

01-june-2018 Added Opt. 356 NF(10.3)

19-Dec-2018 New command

[:SENSe]:FREQuency:IF:STARt?

(Read Only) Query the IF (DUT Out) start frequency.

**Relevant Modes** NF

**Parameters** n/a

**Examples** FREQ:STAR:IF:STAR?

**Query Syntax** :SENSe:FREQuency:IF:STARt?

**Return Type** Numeric

**Default** n/a

Last Modified:

01-june-2018 Added NF mode Opt. 356 (10.3)

[:SENSe]:FREQuency:LO <num>
(Read-Write) Set and query the DUT LO frequency.

Note: The DUT LO frequency is only relevant for frequency converting DUTs (i.e., downconverter, upconverter, and multi-stage converter).

**Relevant Modes**  
NF

**Parameters**

<numeric>  
LO frequency in Hz. Choose a number between the minimum and maximum frequency of the FieldFox.
This command will accept MIN and MAX as arguments.

**Examples**

```
FREQ:LO 10E6
FREQ:LO MAX
```

**Query Syntax**

[:SENSe]:FREQuency:LO?

**Return Type**  
Numeric

**Default**  
--

Last Modified:

01-june-2018  
Added Opt. 356 NF(10.3)

[:SENSe]:FREQuency:RF:STARt?

(Read Only) Query the (DUT In) start frequency.

**Relevant Modes**  
NF

**Parameters**  

n/a

**Examples**

```
FREQ:STAR:RF:STAR?
```

**Query Syntax**

:SENSe:FREQuencyRF:STARt?

**Return Type**  
Numeric

**Default**  
n/a

Last Modified:

01june2018  
Added NF mode Opt. 356 (10.3)

[:SENSe]:FREQuency:SPAN <num>

(Read-Write) Set and query the frequency span of the trace.

**Relevant Modes**  
CAT, NA, NF, SA, CPM, RTSA

**CPM Mode**  
Limited to between 10 Hz and 100 MHz. SPAN is equivalent to
the “channel power integration bandwidth”.

Parameters

<num> Frequency span in Hz. Choose a number between the maximum MINUS the minimum frequency of the FieldFox. This command will accept MIN and MAX as arguments.

Examples

FREQ:SPAN 10E6
FREQ:SPAN MIN

Query Syntax

[:SENSe]:FREQuency:SPAN?

Return Type

Numeric

Default

Maximum MINUS minimum frequency range of the FieldFox.

Last Modified:

01-june-2018 Added NF mode (10.30)
10-june-2016 Added RTSA mode (9.50)
1-April-2014 Added CPM

[:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?

(Read-Only) Query the ratio of the span to resolution bandwidth value.

Relevant Modes

RTSA

Parameters

Range of returned values: 20 (-2.00E+01) to 280 (-2.800E+02)

Examples

FREQ:SPAN:BAND:RAT?

Query Syntax

FREQ:SPAN:BAND:RAT?

Return Type

num

Default

-2.800E+02

Last Modified:

19-sep-2016 Added new RTSA SCPI A.09.50.00.

[:SENSe]:FREQuency:SPAN:FULL

(Write-Only) Set the frequency span to the entire span of the FieldFox.
Relevant Modes SA, RTSA
Parameters None
Examples FREQ:SPAN:FULL
Query Syntax Not Applicable
Default Not Applicable

Last Modified: 10-june-2016 Added RTSA mode (9.50)

[:SENSe]:FREQuency:SPAN:ZERO

(Write-Only) Set the frequency span to 0 Hz. The center frequency is unchanged.

Relevant Modes SA
Parameters None
Examples FREQ:SPAN:ZERO
Query Syntax Not Applicable
Default Not Applicable

Last Modified:

[:SENSe]:FREQuency:STARt <num>

(Read-Write) Set and query the start frequency of the trace.

Relevant Modes CAT, NA, NF, SA, RTSA
Parameters

<num> Start frequency in Hz. Choose a number between the minimum and maximum frequency of the FieldFox.
This command will accept MIN and MAX as arguments.

Examples FREQ:STAR 10E6
FREQ:STAR MIN

Query Syntax [:SENSe]:FREQuency:STARt?
Return Type Numeric
Default Start frequency of the FieldFox

Last Modified:
[:SENSe]:FREQuency:STEP <value>

(Read-Write) Set and query the frequency step size between data points. This is another method to set the number of data points for the measurement.

**Parameters**

- `<value>`: Frequency step size in Hz.

**Examples**

| FREQ:STEP 1e6 |

**Query Syntax**

[:SENSe]:FREQuency:STEP?

**Return Type**

Numeric

**Default**

50 kHz

---

Last modified:

19-Mar-2014  New command (A.07.50)

---

[:SENSe]:FREQuency:STOP <num>

(Read-Write) Set and query the stop frequency of the trace.

**Parameters**

- `<num>`: Stop frequency in Hz. Choose a number between the minimum and maximum frequency of the FieldFox.

This command will accept MIN and MAX as arguments.

**Examples**

| FREQ:STOP 10e6 |
| FREQ:STOP MAX |

**Query Syntax**

[:SENSe]:FREQuency:STOP?

**Return Type**

Numeric

**Default**

Stop frequency of the FieldFox

---

Last Modified:

01-june-2018  Added Opt. 356 NF(10.3)

10-june-2016  Added RTSA mode (9.50)
[:SENSe]:IFFLatness:ALIGnment[:STATe]

This command has been replaced by [:SENSe]:ALIGnment:CHEQ[:STATe]. Learn about superseded commands.

(Read-Write) Set and query the RTSA mode IF flatness amplitude alignment (InstAlign) state.

**Relevant Modes**  IQA, RTSA

**Parameters**

<char>  InstAlign state. Choose from:

- **AUTO** - The alignment process is performed every 300 seconds or when the temperature has changed about 1 degree. It occurs only at the beginning of a sweep and takes about 0.5 seconds to complete.
- **HOLD** - The alignment process is stopped, but the last alignment values are applied to subsequent sweeps.
- **OFF** - The alignment process is NOT performed. Only factory correction values are used.

**Examples**  IFFL:ALIG HOLD

**Query Syntax**  [:SENSe]:IFFLatness:ALIGnment[:STATe]?

**Return Type**  Character

**Default**  AUTO

---

Last Modified:

22nov2017  Updated as a superseded note & added IQA for backwards compatibility.

20-sep-2016  Added RTSA command(A.09.50)

---

[:SENSe]:IFFLatness:ALIGnment[:OBSOLETE]:NOW

This command has been replaced by [:SENSe]:ALIGnment:CHEQ:NOW. Learn about superseded commands.

(Write-only) A IF Flatness alignment is performed once just before the next sweep and applied to subsequent sweeps. If the FieldFox is not sweeping, a single sweep is triggered and alignment is performed.

**Relevant Modes**  IQA, RTSA

**Parameters**  None

**Examples**  IFFL:ALIG:NOW

**Query Syntax**  Not Applicable

**Default**  Not Applicable

---

Last Modified:

22nov2017  Updated as a superseded note & added IQA for backwards compatibility.
[:SENSe]:INTegration:JITTer <num>

(Read-Write) Set and query the Integration mode's maximum allowable jitter setting (Jitter Goal) setting.

See also [:SENSe]:INTegration:MODE, [:SENSe]:INTegration:MAXTime, [:SENSe]:INTegration:WARNing, and [:SENSe]:INTegration:TIME.

Relevant Modes  NF

Parameters

<numeric>  Jitter goal. Choose from:
Minimum -100 dB up to
Maximum 100 dB

Examples  INT:JITT 0.80 dB

Query Syntax  INTegration:JITTer?

Return Type  Numeric

Default  0.50 dB

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:INTegration:MAXTime <num>

(Read-Write) Set and query the Integration mode's maximum time per point setting (Max Time/Pt) setting.

See also [:SENSe]:INTegration:MODE, [:SENSe]:INTegration:JITTER, [:SENSe]:INTegration:WARNing, and [:SENSe]:INTegration:TIME.

Relevant Modes  NF

Parameters

<numeric>  Maximum time per point. Choose from:
Varies, because the setting is coupled to other settings

Examples  INT:MAXT 0.04

Query Syntax  INTegration:MAXTime?

Return Type  Numeric

Default  2.00 s

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)
[:SENSe]:INTegration:MODe <char>

(Read-Write) Set and query the Integration mode setting.

See also [:SENSe]:INTegration:JITTER, [:SENSe]:INTegration:MAXTime, [:SENSe]:INTegration:WARNIng, and [:SENSe]:INTegration:TIMe.

Relevant Modes NF

Parameters
<character> Integration mode. Choose from:

AUTO - The FieldFox Noise Figure application chooses the optimum integration time for each of the noise power measurements needed (i.e., noise source on, noise source off, and receiver reference) to compute the device’s noise figure with the selected amount of measurement jitter in the shortest possible time.

FIXED - Time per point can be set, but other integration SCPIs are not settable.

Examples INT:MOD FIXED

Query Syntax INTegration:MODe?

Return Type Character

Default AUTO

Last Modified:
01june2018 Added NF mode Opt. 356 (10.3)

[:SENSe]:INTegration:TIMe <num>

(Read-Write) Set and query the Integration mode's time per point setting (Time/Pt) setting. Use this command when [:SENSe]:INTegration:MODe = FIXED.

See also [:SENSe]:INTegration:MODe, [:SENSe]:INTegration:JITTER, [:SENSe]:INTegration:WARNIng, and [:SENSe]:INTegration:MAXTime.

Relevant Modes NF

Parameters
<num> Time per point. Choose from:
Varies, because the setting is coupled to other settings

Examples INT:TIM 0.04

Query Syntax INTegration:TIMe?

Return Type Numeric

Default 5.0E-01
[:SENSe]:INTegration:WARNing <bool>

(Read-Write) Set and query the Integration mode's jitter warning (Jitter Warning). Enables and disables the jitter warning when [:SENSe]:INTegration:MODe = AUTO.

See also [:SENSe]:INTegration:MODe, [:SENSe]:INTegration:JITTER, [:SENSe]:INTegration:MAXTime, and [:SENSe]:INTegration:TIMe.

Relevant Modes  NF

Parameters

<boolean> Integration warning per point. Choose from:
- ON(1) - Enables the jitter warning.
- OFF(0) - Disables the jitter warning.

Examples  INT:WARN 1

Query Syntax  INTegration:WARNing?

Return Type  Boolean

Default  1

Last Modified: 01june2018  Added NF mode Opt. 356 (10.3)

[:SENSe]:LTEFdd:CCARrier:LIST<n>:ADD

(Write) Add the current channel and band to a favorite list setup.

Relevant Modes  LTE FDD

Parameters

<n> Setup number. If unspecified, value is set to 1.
- n = 1 | 2 | 3 | 4 | 5 (Up to five favorites can be stored at one time.)

Examples  LTEF:CCAR:LIST3:ADD
          LTEF:CCAR:LIST5:ADD

Query Syntax  n/a

Return Type  n/a

Last Modified: 01june2018  Added NF mode Opt. 356 (10.3)
Default 1

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier:LIST<n>:BAND?

(Read Only) Query the current band in the specified favorites setup.

Relevant Modes LTE FDD

Parameters

<n> Setup number. If unspecified, value is set to 1.
\[n = 1 \mid 2 \mid 3 \mid 4 \mid 5\] (Up to five favorites can be stored at one time.)

<numeric>

Examples no write command

Query Syntax LTEF:CCAR:LIST3:BAND?

Return Type numeric

Default 1

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier:LIST<n>:CHANnel?

(Read Only) Query the current channel in the specified favorites setup.

Relevant Modes LTE FDD

Parameters

<n> Setup number. If unspecified, value is set to 1.
\[n = 1 \mid 2 \mid 3 \mid 4 \mid 5\] (Up to five favorites can be stored at one time.)

<numeric>

Examples no write command

Query Syntax LTEF:CCAR:LIST3:CHAN?
Return Type numeric
Default 1

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier:LIST<n>:REMove

(Write) Disable/clear the current channel and band favorites setup number.

Relevant Modes LTE FDD

Parameters

<n> Setup number. If unspecified, value is set to 1.
   n = 1 | 2 | 3 | 4 | 5 (Up to five favorites can be stored at one time.)

Examples LTEF:CCAR:LIST3:REM
            LTEF:CCAR:LIST5:REM

Query Syntax n/a
Return Type n/a
Default 1

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier<n>:BAND <num>

(Read-Write) Set and query the component carrier band (BAND). If Band is changed, Channel is updated also. The rule is defined in 3GPP TS 36.101.

If FreqMode = Channel, center frequency is updated based on band and channel in target carrier.

Relevant Modes LTE FDD

Parameters

<n> Component carrier (CC) number. If unspecified, value is set to 1 (CC0).
   n = 1 | 2 | 3 | 4 | 5 (Up to five component carriers can be measured at one time.)
[:SENSe]:LTEFdd:CCARrier[n]:CHANnel

(Read-Write) Set and query the component carrier channel (CHAN). If the channel is changed, the band value is updated also. The rule is defined in 3GPP TS 36.101.

If FreqMode = Channel, center frequency is updated based on band and channel in target carrier.

Relevant Modes  LTE FDD

Parameters

<n> Component carrier (CC) number. If unspecified, value is set to 1 (CC0).

n = 1 | 2 | 3 | 4 | 5 (Up to five component carriers can be measured at one time.)

<num>

Examples

| LTEF:CCAR5:CHAN 3800 |
| LTEF:CCAR3:CHAN 4149 |

Query Syntax  LTEF:CCAR:CHAN?

Return Type  number

Default  0

Last Modified:

19-Dec-2018  New command
(Read-Write) Set and query the component carrier's enable/disable state.

**Relevant Modes** LTE FDD

**Parameters**

<n> Component carrier (CC) number. If unspecified, value is set to 1 (CC0).

\( n = 1 \mid 2 \mid 3 \mid 4 \mid 5 \) (Up to five component carriers can be measured at one time.)

<numeric> Choices:

| ON (1): Enables the current component carrier. |
| OFF (0): Disables the current component carrier. |

**Examples**

- `LTEF:CCAR5:ENAB ON`
- `LTEF:CCAR3:ENAB 0`

**Query Syntax** `LTEF:CCAR:ENAB?`

**Return Type** number

**Default** 1000000000

---

Last Modified:

19-Dec-2018 New command

---

[:SENse]:LTEFdd:CCARier<n>:FREQuency:CENTer <num>

(Read-Write) Set and query the component carrier center frequency.

**Relevant Modes** LTE FDD

**Parameters**

<n> Component carrier (CC) number. If unspecified, value is set to 1 (CC0).

\( n = 1 \mid 2 \mid 3 \mid 4 \mid 5 \) (Up to five component carriers can be measured at one time.)

<numeric> Choices:

- Maximum (Hz): Dynamic. Varies with the FieldFox model.
- Minimum: 0 Hz

**Examples**

- `LTEF:CCAR5:FREQ:CENT 2e9`
- `LTEF:CCAR3:FREQ:CENT 10000`

**Query Syntax** `LTEF:CCAR:FREQ:CENT?`

**Return Type** number

**Default** 1000000000

---

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier<n>:LIST<x>:APPLY

(Write) Apply a saved favorite setup channel and band to the FieldFox.

Relevant Modes LTE FDD

Parameters

<n> Component carrier (CC) number. If unspecified, value is set to 1 (CC0).
   \( n = 1 \mid 2 \mid 3 \mid 4 \mid 5 \) (Up to five component carriers can be measured at one time.)

<x> Setup number. If unspecified, value is set to 1.
   \( x = 1 \mid 2 \mid 3 \mid 4 \mid 5 \) (Up to five favorites can be available.)

Examples LTEF:CCAR2:LIST3:APP

Query Syntax n/a

Return Type n/a

Default \( n=1, \ x=1 \)

Last Modified:
19-Dec-2018 New command

[:SENSe]:LTEFdd:CCARrier<n>:MODE <string>

(Read-Write) Set and query the LTE FDD frequency mode (CHANnel | FREQuency).

Relevant Modes LTE FDD

Parameters

<string> Choices:

CHANnel: Channel and Band are based on the component carrier in the FieldFox. Center frequency is coupled to the channel and band choices.

FREQuency: Center frequency is entered independently.

If FreqMode = Channel, center frequency value is based on band and channel in target carrier.

Examples LTEF:CCAR2:MODE CHAN

LTEF:CCAR2:MODE FREQ
Query Syntax  LTEF:CCAR5:MODE?
Return Type  string
Default  FREQ

Last Modified:
19-Dec-2018  New command

[:SENSe]:LTEFdd:FERRor:THReshold <num>
(Read-Write) Set and query frequency error threshold value.

Relevant Modes  LTE FDD
Parameters
<numeric>  Frequency error value in Hz.
Examples  LTEF:FERR:THR 3000
Query Syntax  :LTEF:FERR:THR?
Return Type  Numeric
Default  900 Hz

Last Modified:
19-Dec-2018  New command

[:SENSe]:LTEFdd:FLRange:EXTended <bool>
(Read-Write) Set and query the extended frequency range.

Relevant Modes  LTE FDD
Parameters
<boolean>  ON (1): Enable the extended frequency range.
            OFF (2): Disable the extended frequency range.
Examples  LTEF:FLR:EXT ON
Query Syntax  :LTEF:FLR:EXT?
Return Type  numeric
Default  OFF (0)
[:SENSe]:MEASure

(Read-Write) Selects between Density, Spectrogram, and Real-Time traces.

**Relevant Modes** RTSA

**Parameters**

<char> Select the type of Real-Time measurement to be displayed:

- **DENSity** - Signal is displayed is a real-time spectrum with a histogram bitmap.
- **SPECTrogram** - The signal is displayed is a real-time spectrum with a spectrogram bitmap. Spectrogram views are valuable in capturing signals that include both time and frequency variations.
- **NORMAL** - The signal is displayed as a real-time spectrum. Use real-time spectrogram to quickly display measurements.

**Examples** MEAS DENS

**Query Syntax** [:SENSe]:MEASure?

**Return Type** Character

**Default** DENS

---

Last Modified:

19-sep-2016 Added new RTSA command (A.09.50).

[:SENSe]:MEASure:SPERiod <num>

(Read-Write) Set and query the sample period.

**Relevant Modes** IQA

**Parameters**

<num> Minimum: 80 ns
Maximum: 80 ms
Couplings: Sample Period = (1 / Sample Rate)

**Examples** MEAS:SPER 300e-9

**Query Syntax** MEAS:SPER?

**Return Type** Numeric

**Default** 8.0000000E-08

---

Last Modified:
::SENSe::MEASure:SRATe <num>

(Read-Write) Set and query the IQA sample rate. The “Digital IF Bandwidth” parameter controls the resampler section of the hardware. It is directly coupled to the sample rate.

Sample Rate = (Digital IF Bandwidth) * (Over Sample Ratio)
= (Digital IF Bandwidth) * 1.25

Relevant Modes  IQA

Parameters
<numeric> Minimum: 12.5 Hz
Maximum: 12.5 MHz
Couplings: (1.25 * IF Bandwidth)

Examples

MEAS:SRAT 1 MHz
MEAS:SRAT 10000

Query Syntax MEAS:SRAT?
Return Type Numeric
Default 1.250000000E+07

Last Modified:
22-Oct-2017 Added IQA mode (10.1x)

::SENSe::MEASurement:ADEMod

(Read-Write) Set the AM/FM Metrics measurement type (Select 1 of n measurement choices)

Relevant Modes  SA

Parameters  NONE - Turns off the AM/FM functionality.
LAM - Tune & Listen AM
LFMN - Tune & Listen narrow band FM
LFMW - Tune & Listen wide band FM
MAM - AM Metrics
MFM - FM Metrics

<string>

Examples  :MEAS:ADEM NONE

Query Syntax :MEAS:ADEM?
Return Type  String
Default MAM
[:SENSe]:MEASurement:AOFF

(Write-Only) Turns OFF the currently-displayed Channel or Interference Analysis measurements. Reverts to standard SA display.

Relevant Modes:  SA
Parameters:  None
Examples:  MEAS:AOFF
Query Syntax:  Not Applicable
Default:  Not Applicable

[:SENSe]:MEASure:CAPTure:LENGth

(Read-Write) Set and query the IQA data capture length via the sample number.
Note: This capture length sample value is only used with a captured IQ data file.
Use SENS:MEAS:CAPT:TIME to set the data capture time for the IQA data.

Relevant Modes:  IQA
Parameters:  
<captureLength>  Data IQA capture capture.
  Minimum: 1
  Maximum: 4194304
  Couplings:  Capture Time. Changes will effect Capture Time
  (Capture Samples = Capture Time * Sample Rate + 1).
Examples:  MEAS:IQC:CAPT:TIME  6e-6
Query Syntax:  MEAS:IQC:CAPT:TIME?
Return Type:  integer
Default:  1251

Last Modified:

16-Dec-2015  Added AM/FM Metrics (8.15 & 9.15)
22-oct-2017  Added new IQA mode content (10.1x).
### [:SENSe]:MEASure:CAPTure:TIME

(Read-Write) Set and query the filename for I/Q data capture time.

**Note:** The Maximum value of this parameter is limited by the maximum capture samples that is available for the different platforms and determined by the current sample rate.

Use `MMEM:STOR:IQC:FTYP` to set the data capture file type.

**Relevant Modes**  
IQA

**Parameters**

- `<time>`  
  Data I/Q capture time.  
  **Minimum:** 1 ps  
  **Maximum:** Dynamic change. In the default setting, Sample Rate = 12.5 MHz, the max value is 335.544ms.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>MEAS:IQC:CAPT:TIME 6e-6</th>
</tr>
</thead>
</table>

**Query Syntax**

`MEAS:IQC:CAPT:TIME?`

**Return Type**  
numeric

**Default**  
100 s

---

Last Modified:

22-oct-2017    Added new IQA mode content (10.1x).

---

### [:SENSe]:MEASurement:CHANnel <char>

(Read-Write) Set and query the current SA mode channel measurement.

**Relevant Modes**  
SA

**Parameters**

- `<char>`  
  Channel measurement. Choose from:  
  - **CHP** - Channel Power  
  - **OBW** - Occupied Bandwidth  
  - **ACPR** - Adjacent Channel Power  
  - **NONE** - No current channel measurement.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>MEAS:CHAN ACPR</th>
</tr>
</thead>
</table>

**Query Syntax**

[:SENSe]:MEASurement:CHANnel?

**Return Type**  
Character

**Default**  
NONE

---

**SENSe:MEASurement:ERTA:PNID**
(Read-Write) Set and query the partner IP address.

Relevant Modes  ERTA

Parameters

<string> Network identity of the partner (source) FieldFox, enclosed in quotes. Currently we recommend using the IP address. But in the future, we may accept a hostname string.

Examples  MEAS:ERTA:PNID "192.168.0.0"

Query Syntax [:SENSe]:MEASurement:ERTA:PNID?

Return Type  String

Default  Not applicable

Last Modified:
4-May-2015  New command (8.04)

[:SENSe]:MEASurement:ERTA:PSTatus

(Read-Write) Set and read Partnership status. Can only set Master (Receiver).

Relevant Modes  ERTA

Parameters

<char> Receiver status. Choose from:
OFF - No partnership exists.
MAST - Master status

Examples  MEAS:ERTA:PST OFF

Query Syntax [:SENSe]:MEASurement:ERTA:PSTatus?

Return Type  Character

Default  OFF

Last Modified:
20-Jan-2015  New command (8.00)

[:SENSe]:MEASurement:ERTA:PVERify?

(Read-only) Returns whether or not the partner (source) FieldFox is ERTA network capable.

Relevant Modes  ERTA
Parameters  None
Return Type  Boolean
  1 - Partner is ERTA capable.
  0 - Partner is NOT ERTA capable
Default  Not applicable

Last Modified:
4-May-2015  New command (8.04)

[:SENSe]:MEASurement:ERTA:ROLE?
(Read-only) Read the ERTA stimulus-response role.

Relevant Modes  ERTA
Parameters  None
Return Type  Character
  NONE - No partnership exists.
  SRC - FieldFox is a Source (follower).
  RCVR - FieldFox is a Receiver (master)
Default  None

Last Modified:
20-Jan-2015  New command (8.00)

[:SENSe]:MEASurement:INTerference <char>
(Read-Write) Set and query the current SA mode Interference Analysis (Opt 236) display.

Relevant Modes  SA
Parameters
  <char>  Interference Analysis display. Choose from:
    SPECtrogram
    WATerfall
    NONE - no Interference Analysis measurement displayed.
Examples  MEAS: INT SPEC
Query Syntax  [:SENSe]:MEASurement:INTerference?
Return Type  Character
[:SENSe]:MEASurement:PRESet

(Write-Only) Resets the currently-displayed channel measurement to its default settings. The Center Frequency, Preamp ON|OFF, RF Attenuation, Markers, Limits, and Radio Standard settings are NOT reset.

Relevant Modes SA
Parameters None
Examples MEAS : PRES
Query Syntax Not Applicable
Default None

[:SENSe]:MEAS:TALlisten <char>

(Read-Write) Set and query the Tune and Listen demodulation type for SA Mode.

Relevant Modes SA
Parameters None
Examples MEAS : TAL FMW
Query Syntax [:SENSe]:MEAS:TALlisten?
Return Type Character
Default NONE

[:SENSe]:NBANdwidth <num> or <char>

(Read-Write) Set and query the NF resolution bandwidth for the measurement.

Relevant Modes NF
Parameters None
Examples NBAN 1e3
NBAN MAX
NBAN MIN
Query Syntax: `SENSe:NBANdwidth?`

Return Type: Numeric

Default: 5 MHz for NF

Query Syntax: `[:SENSe]:OBW:PPOW <num>`

(Read-Write) Set and query the Power Percent for an Occupied Bandwidth measurement.

Relevant Modes: SA

Parameters: `<num>` Power Percent. Choose a number from 10 to 99.99.

Examples: `OBW:PPOW 20`

Query Syntax: `[:SENSe]:OBW:PPOW?`

Return Type: Numeric

Default: 10

[:SENSe]:POINt:DWELl <value>

(Read-Write) Set and query the settling time after the internal source steps to the next frequency and before the power sensor makes a measurement.

Relevant Modes: FOPS (Opt 208)

Parameters: `<value>` Dwell time in seconds.

Examples: `POIN:DWEL .01`

Query Syntax: `[:SENSe]:POINt:DWELl?`

Return Type: Numeric

Default: 0

Last modified:

01-june-2018 Added Opt. 356 NF(10.3)

19-Mar-2014 New command (A.07.50)
(Read-Write) Set and query the maximum number of readings the power sensor will make to achieve settling.

Each power sensor reading is "settled" when either:

- two consecutive readings are within the Tolerance value (see [:SENSe]:TOL) or
- when the Max Number of Readings has been met.

The readings that were taken are averaged together to become the "settled" reading.

Relevant Modes  **FOPS** (Opt 208)

Parameters

<value>  Max number of readings. Choose a value between 1 and 20

Examples  POIN:READ:MAX  5

Query Syntax  [:SENSe]:POINt:READ:MAX?

Return Type  Numeric

Default  4

Last modified:  
19-Mar-2014  New command (A.07.50)

[:SENSe]:POWer[:RF]:ATTenuation <num>

(Read-Write) Set and query RF attenuation value. Also set **POW:ATT:AUTO OFF**.

Relevant Modes  **5GTF, SA, LTE FDD, CPM, RTSA, IQA**

Parameters

<num>  Attenuation value in dB. Choose a number between the minimum and maximum attenuation of the FieldFox.  
This command will accept MIN and MAX as arguments.  
The RF Attenuation value is never automatically lowered below 5 dB.  
CPM mode only accepts two values: 10 dB or 30 dB

Examples  POW:ATT  30

Query Syntax  :POWer:ATTenuation?

Return Type  Numeric

Default  10

Last Modified:  
03dec2018  Added LTE FDD & 5GTF (A.11.0)
[:SENSe]:POWer[:RF]:ATTenuation:AUTO <bool>

(Read-Write) Set and query the Auto RF Attenuation state. See also :POW:ATT.

**Relevant Modes** 5GTF, SA, LTE FDD, RTSA, IQA

**Parameters**

<boolean> Auto RF Attenuation state. Choose from:

- **O** or **OFF** - RF Attenuation is set manually
- **1** or **ON** - RF Attenuation is set automatically

**Examples**

POW:ATT:AUTO 1

**Query Syntax** [:SENSe]:POWer[:RF]:ATTenuation:AUTO?

**Return Type** Boolean

**Default** ON

---

Last Modified:

- 03dec2018 Added LTE FDD & 5GTF (A.11.0)
- 22oct2017 Added IQA mode (10.1x)
- 10-june-2016 Added RTSA mode (9.50)

[:SENSe]:POWer:BPLevel <num>

(Read-Write) Set and query the Blue power level limit used with Bar Chart displays (Option LTE FDD & 5GTF only).

See also: [:SENSe]:POWer:RPLevel (Red power level)

**Relevant Modes** LTE FDD, 5GTF

**Parameters**

<num> Select the LOWEST power levels (in dB) that you expect to measure. This power level, and those measured lower than this level, appear Blue.

- Enter a value between the 0 and the current Red (Highest) limit. (Default Red is -50)
- Couplings: If it’s over range with max value, Blue limit = Red limit -1

**Examples** POW:BPL -80
### [:SENSe]:POWer[:RF]:EXTGain <num>

**Query Syntax**  
[:SENSe]:POWer[:RF]:EXTGain?

**Return Type**  
Numeric

**Default**  
-90

**Last Modified:**  
19-Dec-2018 Adding new command for LTE FDD & 5GTF (A.11.0)

**Purpose:**  
Set and query external gain value. Use to compensate the trace and scale when using an external amp or attenuator. May automatically increase the amount of attenuation.

**Relevant Modes**  
5GTF, SA, LTE FDD, RTSA, IQA

**Parameters**  
- <numeric>  
  Amount of gain (positive value) or loss (negative value) in dB. This command will accept MIN and MAX as arguments.
  - Minimum: -100 dB
  - Maximum: 100 dB

**Examples**  
POW:EXTG 10

#### Last Modified:

- 03dec2018 Added LTE FDD & 5GTF (A.11.0)
- 22oct2017 Added IQA mode (10.1x)
- 10-june-2016 Added RTSA mode (9.50)

### [:SENSe]:POWer[:RF]:GAIN:AUTO <bool>

**Query Syntax**  
[:SENSe]:POWer[:RF]:GAIN:AUTO?

**Return Type**  
Numeric

**Default**  
0

**Last Modified:**  
19-Dec-2018 Adding new command for LTE FDD & 5GTF (A.11.0)

**Purpose:**  
Set and query preamplifier state. Available with Option 235.

**Relevant Modes**  
IQA

**Parameters**  
- <boolean>  
  Preamp AUTO ON | OFF state. Choose from:
  - O or OFF - AUTO Preamp OFF
  - 1 or ON - AUTO Preamp ON

When Auto is enabled, the preamp setting is coupled to the RF attenuation.
and to the reference level. When the reference level reaches a certain value, the preamp is turned on and **PA** is shown to the left of the graticules on the display.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POW:GAIN:AUTO 0</td>
</tr>
<tr>
<td>POW:GAIN:AUTO OFF</td>
</tr>
</tbody>
</table>

**Query Syntax**

`:POW:GAIN:AUTO?`

**Return Type**

Boolean

**Default**

OFF (0)

---

**Last Modified:**

22oct2017 Added IQA mode (10.1x)

---

### [:SENSe]:POWer[:RF]:GAIN[:STATe] <bool>

*(Read-Write)* Set and query preamplifier state. Available with Option 235.

**Relevant Modes**

5GTF, SA, LTE FDD, RTSA, IQA

**Parameters**

<boolean> Preamp ON | OFF state. Choose from:
- **0** or **OFF** - *(For 5GTF and LTE FDD only OFF is valid)* Preamp OFF
- **1** or **ON** - *(For 5GTF and LTE FDD only ON is valid)* Preamp ON
- **AUTO** - *(5GTF, LTE FDD, and RTSA Only!)* When Auto is selected, the preamp setting is coupled to the RF attenuation and to the reference level. When the reference level reaches a certain value, the preamp is turned on and **PA** is shown to the left of the graticules on the display.

For IQA, see also **POW:GAIN:AUTO**.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>POW:GAIN 0</td>
<td>'0 and 1 are invalid choices for 5GTF and LTE FDD</td>
</tr>
<tr>
<td>POW:GAIN AUTO</td>
<td>'5GTF, LTE FDD, and RTSA Only</td>
</tr>
</tbody>
</table>

**Query Syntax**

[:SENSe]:POWer[:RF]:GAIN[:STATe]?

**Return Type**

Boolean

**Default**

OFF

---

**Last Modified:**

03dec2018 Added LTE FDD & 5GTF (A.11.0)

22oct2017 Added IQA mode (10.1x)

10-june-2016 Added RTSA mode (9.50)
[:SENSe]:POWer[:RF]:RLEVel <num>

(Read-Write) Sets the absolute power reference level.

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

- `<num>` Absolute power reference level in dBm
  - Preset: -10 dBm
  - Minimum: -210 dBm
  - Maximum: 30 dBm

**Couplings**  N/A

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>POW:RLEV 10</td>
</tr>
<tr>
<td>POW:RLEV -23</td>
</tr>
</tbody>
</table>

**Query Syntax**  POWer:RLEVel?

**Default**  -10 dBm

---

Last Modified:

- 03dec2018  Added LTE FDD & 5GTF (A.11.0)
- 19-Dec-2018  New command

---

[:SENSe]:POWer:RPLevel <num>

(Read-Write) Set and query the Red power level limit used with Bar Chart displays (LTE FDD and 5GTF only).

See also: [:SENSe]:SPECtrogram:BPLevel (Blue power level)

**Relevant Modes**  LTE FDD, 5GTF

**Parameters**

- `<num>` Select the HIGHEST power levels (in dB) that you expect to measure. This power level, and those measured higher than this level, appear Red.
  - Enter a value between 0 and the current Blue (Lowest) limit. (Default Blue is -90).
  - Couplings: If it’s over range with min value, Red limit = blue limit + 1

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>POW:RPL -10</td>
</tr>
</tbody>
</table>

**Query Syntax**  [:SENSe]:POWer:RPLevel?

**Return Type**  Numeric

**Default**  -50

---

Last Modified:
[:SENSe]:QUANtity:TACTive?

(Read-only) Returns the number of active traces and the number of traces that are currently allowed. Some SA measurements allow only one measurement trace, such as Channel Mode measurements. In this case, 1,1 would be returned.

Use TRACe<n>:DATA? to read SA trace data.
Use TRACe<n>:TYPE to set the trace type.

Relevant Modes  SA
Parameters
Examples  QUAN:TACT?
'With a general SA measurement active, returns 1,4
'With a Channel power measurement active, returns 1,1

Return Type  Numeric, Numeric
Default  1,4

Last Modified:
20-Oct-2010  New command (5.30)

[:SENSe]:RADio:CHANnel:CENTer <num>

(Read-Write) Set and query the channel number that appears in the center of the current channel measurement. First select a Radio Standard [:SENSe]:RADio:STANdard then activate the channel frequency-drive commands by changing [:SENSe]:RADio:TEUNit to CHAN.

Relevant Modes  SA, CPM
Parameters
<num>  Channel number. The range of valid channel numbers depends on the current radio standard. If a channel number outside the acceptable range is entered, the channel will be set to the closest valid channel number.

Examples  RAD:CHAN:CENT 10

Query Syntax  [:SENSe]:RADio:CHANnel:CENTer?
Return Type  Numeric
Default  Center channel of the radio standard.
[:SENSe]:RADio:CHANnel:DIRection <char>

(Read-Write) Set and query the frequency Uplink or Downlink direction. First select a Radio Standard [:SENSe]:RADio:STANdard then activate the channel frequency-drive commands by changing [:SENSe]:RADio:TEU nit to CHAN.

NOT all Radio Standards contain Uplink and Downlink frequencies.

Relevant Modes SA, CPM

Parameters

<char> Choose from:
- UP - Uplink
- DOWN - Downlink

Examples RAD:CHAN:DIR UP

Query Syntax [:SENSe]:RADio:CHANnel:DIRection?

Return Type Character

Default DOWN

Last Modified: 1-Apr-2014 Added CPM

[:SENSe]:RADio:CHANnel:STARt <num>

(Read-Write) Set and query the channel number that appears at the start (left edge) of the current channel measurement. First select a Radio Standard [:SENSe]:RADio:STANdard then activate the channel frequency-drive commands by changing [:SENSe]:RADio:TEU nit to CHAN.

Relevant Modes SA

Parameters

<num> Channel number. The range of valid channel numbers depends on the current radio standard. If a channel number outside the acceptable range is entered, the channel will be set to the closest valid channel number.

Examples RAD:CHAN:STAR 10

Query Syntax [:SENSe]:RADio:CHANnel:STARt?

Return Type Numeric

Default -1 (Start and Stop channels not activated)
[:SENSe]:RADio:CHANnel:STEP <num>

(Read-Write) Set and query the channel number step size which determines the number of channels that are incremented and decremented each time the Up/Down arrows are pressed on the FieldFox front-panel.

First select a Radio Standard [:SENSe]:RADio:STANdard then activate the channel frequency-drive commands by changing [:SENSe]:RADio:TEUNit to CHAN.

Relevant Modes: SA, CPM

Parameters

<num> Channel number step size. Choose a value between 1 and 100.

Examples: RAD:CHAN:STEP 10

Query Syntax: [:SENSe]:RADio:CHANnel:STEP?

Return Type: Numeric

Default: 1

Last Modified:
1-Apr-2014 Added CPM

[:SENSe]:RADio:CHANnel:STOP <num>

(Read-Write) Set and query the channel number that appears at the stop (right edge) of the current channel measurement. First select a Radio Standard [:SENSe]:RADio:STANdard then activate the channel frequency-drive commands by changing [:SENSe]:RADio:TEUNit to CHAN.

Relevant Modes: SA

Parameters

<num> Stop channel number. The range of valid channel numbers depends on the current radio standard. If a channel number outside the acceptable range is entered, the channel will be set to the closest valid channel number.

Examples: RAD:CHAN:STOP 10

Query Syntax: [:SENSe]:RADio:CHANnel:STOP?

Return Type: Numeric

Default: -1 (Start and Stop channels not activated)

[:SENSe]:RADio[:SELect]:STANdard <char> - Obsolete

The command is replaced with [:SENSe]:RADio:STANdard[:SELect] which uses a string argument.

(Read-Write) Set and query the Radio Standard for use in channel measurements.

Relevant Modes: SA
Parameters

<char> Radio Standard. Not case-sensitive. Choose from:
- NONE
- GSM450
- GSM480
- GSM850
- GSM900
- EGSM900
- RGS900 // more band extension beyond E-GSM 900
- GSM1800
- GSM1900
- IDEN800
- IDEN900
- WCDMA850
- WCDMA1900
- WCDMA2100
- WCDMAAWS
- TDSCDMACHINA
- CDMA2K850
- CDMA2K1900
- CDMA2KJAPAN
- CDMA2KKOREA
- LTE700US // E-UTRA band 13 for 700M US LTE
- LTE2600 // E-UTRA band 7 for 2600M International LTE
- WIMAXMOBILE
- DVBT"HF
- DVBTUHF

Examples
radio:standard gsm1800

Query Syntax
[:SENSe]:RADio[:SELect]:STANdard?

Return Type Character

Default None

[:SENSe]:RADio:STANdard[:SELect] <string>

This command replaces [:SENSe]:RADio:STANdard.
(Read-Write) Set and query the Radio Standard for use in channel measurements.

Relevant Modes SA, CPM

Parameters
Radio Standard. Choose from those listed in the User Interface when the Radio Standard softkey is pressed, including custom radio standards. **Case-sensitive.** Include spaces and enclose in double-quotes. Select "None" to turn radio standards OFF.

**Examples**  
radio:standard "GSM 1800"

**Query Syntax**  
[:SENSe]:RADio:STANdard[:SELect]?

**Return Type**  
String

**Default**  
None

Last Modified:
1-Apr-2014 Added CPM
11-Aug-2011 New command

**[:SENSe]:RADio:TEUNit <char>**

*(Read-Write)* Set and query the whether channel measurements are tuned using frequency or channel numbers.

**Relevant Modes**  
SA, CPM

**Parameters**

<num>  
Tune entry units. Choose from:

FREQ - Channel measurement tuning is accomplished using Frequency (*Sens:*Freq:Start, Stop, Center, Span)

CHAN - Channel measurement tuning is accomplished using Channel numbers (*Sens:*Rad:Chan:Start, Stop, Center). Must also select a Radio Standard [:SENSe]:RADio:STANdard.

**Examples**  
RAD:TEUN CHAN

**Query Syntax**  
[:SENSe]:RADio:TEUNit?

**Return Type**  
Character

**Default**  
FREQ

Last Modified:
1-Apr-2014 Added CPM

**[:SENSe]:RESolution <char>**

*(Read-Write)* Set and query the number of data points for the Pulse measurement.

**Relevant Modes**  
Pulse Measurements
Parameters
<char> Resolution setting. Choose from the following:
LOW - 240 data points.
MED - 1000 data points.
HIGH - Calculated value - approximately 8000 data points.

Examples
RES LOW

Query Syntax [:SENSe]:RESolution?

Return Type Character

Default LOW

Last Modified: 29-Oct-2013 New command

[:SENSe]:ROSCillator:SOURce <char>

(Read-Write) Set and query the source of the 10 MHz reference oscillator.

Relevant Modes ALL

Parameters
<num> Choose from:
INTernal - Internal FieldFox reference.
EXTernal - An external reference.

Examples ROSC:SOUR INT

Query Syntax [:SENSe]:ROSCillator:SOURce?

Return Type Character

Default INTernal

[:SENSe]:ROSCillator:STATus?

(Read-Only) Returns the status of the reference loop.
-1 External Reference Unlocked
0 Internal Reference
+1 External Reference Locked

Relevant Modes ALL

Examples ROSC:STAT?

Return Type Numeric

Default Not Applicable
[:SENSe]:SPECtrogram:BPLevel <num>

(Read-Write) Set and query the Blue power level limit used with Waterfall (SA Only) and Spectrogram displays (For SA mode, refer to Interference Analyzer 0pt 236).

See also: [:SENSe]:SPECtrogram:RPLevel (Red power level)

Relevant Modes  SA, RTSA

Parameters

<num> Select the LOWEST power levels (in dB) that you expect to measure. This power level, and those measured lower than this level, appear Blue.

Enter a value between the 0 and the current Red (Highest) limit. (Default Red is -50)

Examples  SPEC:BPL -80

Query Syntax  [:SENSe]:SPECtrogram:BPLevel?

Return Type  Numeric

Default  -90

Last Modified:

19-sep-2016  Added  RTSA as a relevant mode (A.09.50).

[:SENSe]:SPECtrogram:RPLevel <num>

(Read-Write) Set and query the Red power level limit used with Waterfall (SA Only) and Spectrogram displays (For SA mode, refer to Interference Analyzer 0pt 236).

See also: [:SENSe]:SPECtrogram:BPLevel (Blue power level)

Relevant Modes  SA, RTSA

Parameters

<num> Select the HIGHEST power levels (in dB) that you expect to measure. This power level, and those measured higher than this level, appear Red.

Enter a value between 0 and the current Blue (Lowest) limit. (Default Blue is -90)

Examples  SPEC:BPL -10

Query Syntax  [:SENSe]:SPECtrogram:RPLevel?

Return Type  Numeric

Default  -50

Last Modified:

19-sep-2016  Added  RTSA as a relevant mode (A.09.50).
[:SENSe]:SPECTrogram:TMARker:STATe <char>

(Read-Write) Set and query the state of the time and delta markers used with Waterfall and Spectrogram displays (Interference Analyzer 0pt 236).

A Time marker alone displays the time from the first record to the location of the time marker.

A Delta marker display the difference between the time marker and the delta marker.

Spectrogram displays the most recent record at the top of the screen. Waterfall displays the most recent record at the bottom.

Use [:SENSe]:SPECTrogram:TMARker:VALue to move the Time and Delta markers.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA</th>
</tr>
</thead>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>&lt;num&gt;</th>
<th>Marker state. Choose from:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>Time and Delta markers OFF</td>
</tr>
<tr>
<td>DELT</td>
<td>Delta markers enabled.</td>
</tr>
<tr>
<td>TIME</td>
<td>Time marker enabled.</td>
</tr>
</tbody>
</table>

**Examples**

SPEC:TMAR:STAT DELT

**Query Syntax**

[:SENSe]:SPECTrogram:TMARker:STATe?

**Return Type** Character

**Default** OFF

---

Last Modified:

24-Jan-2012 New command

[:SENSe]:SPECTrogram:TMARker:VALue <num>

(Read-Write) Set and query the location of the Time or Delta marker, used with Waterfall and Spectrogram displays (Interference Analyzer 0pt 236).

Use [:SENSe]:SPECTrogram:TMARker:STATe to enable the Time or Delta marker.

Spectrogram displays the most recent record at the top of the screen. Waterfall displays the most recent record at the bottom.

There is currently no command to read the Time that is displayed on the screen.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA</th>
</tr>
</thead>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>&lt;num&gt;</th>
<th>Time or Delta marker location. Enter a value between 0 (bottom of screen) and the maximum number of records visible on the screen. The max number depends on the Waterfall view and Spectrogram Angle settings.</th>
</tr>
</thead>
</table>

**Examples**

SPEC:TMAR:VAL 10

**Query Syntax**

[:SENSe]:SPECTrogram:TMARker:VALue?

**Return Type** Numeric
Default 0

Last Modified:
17-Aug-2012 Edited for both marker types

[:SENSe]:SPECtrogram:TPDivision

(Read-Write) Set and query the Time per division of the Spectrogram trace graph window.

Relevant Modes  RTSA

Parameters
<num> Per division time in seconds (0.0005 seconds to 10 seconds. Default = 100 ms).

Examples SPEC:TPD 1e-3
     SPEC:TPD 5

Query Syntax [:SENSe]:TIME:PDIVision?

Return Type Numeric

Default 1.000E-01

Last Modified:
19-sep-2016 New RTSA mode command (A.09.50)

[:SENSe]:SPECtrogram:VIEW <char>

(Read-Write) Set and query the Spectrogram View setting, which changes how the spectrogram is displayed relative to the data trace. This setting affects the total number of trace records that can be displayed.(Interference Analyzer 0pt 236).

Relevant Modes  SA, RTSA

Parameters
<num> Spectrogram view. Choose from:

   OVERlay - Displays the data trace over the spectrogram with NO graticules. Total trace records: 340.
   FULL  - Displays the spectrogram with NO data trace. Total trace records: 340.
   TOP   - Displays the data trace above the spectrogram. Total trace records: 130.
   BOTTOM - Displays the data trace below the spectrogram. Total trace records: 130.
Examples | SPEC:VIEW TOP
--- | ---
Query Syntax | [:SENSe]:SPECTrogram:VIEW?
Return Type | Character
Default | OVERlay

Last Modified:
19-sep-2016 Added RTSA as a relevant mode (A.09.50).

[SENSe]:SPECTrogram:WANGle <char>

(Read-Write) Set and query the Waterfall Angle setting, which allows you to show more or less detail. More detail provides fewer (time) records. (Interference Analyzer 0pt 236).

Relevant Modes | SA
Parameters
<char> Choose from:
MOD - (Moderate) Compromise between detail and number of records.
STEP - Least detail with highest number of records.
GRAD - (Gradual) Most detail with lowest number of records.
WIDE - (Wide Angle) Moderate angle with increased space between records.
Examples | SPEC:WANG WIDE
Query Syntax | [:SENSe]:SPECTrogram:WANGle?
Return Type | Character
Default | MOD

[SENSe]:SPECTrum:BANDwidth[:RESolution] <num>

(Read-Write) Set and query the Resolution Bandwidth setting. Changing this value changes the FFT Window size, Analysis length and the sweep time (measurement capture length).

Relevant Modes | IQA
Parameters
<num> Minimum: 0.1 Hz
Maximum: 3 MHz
Examples | SPEC:BAND 100
Query Syntax | :SPEC:BAND?
Return Type | Character
Default | 2.00000000E+05
[:SENSe]:SPEctrum:BANDwidth[:RESolution]:AUTO <bool>

(Read-Write) Set and query the Resolution Bandwidth setting. This is the resolution bandwidth of the FFT analysis. Enables or disables the FieldFox’s automatic resolution bandwidth setting.

See also, If the SPEC:BAND.

Relevant Modes IQA

Parameters

<boolean> ON (1): Enables the automatic resolution bandwidth setting.
OFF (0): Disables the automatic resolution bandwidth.

Examples

SPEC:BAND:AUTO 0
SPEC:BAND:AUTO ON

Query Syntax :SPEC:BAND:AUTO?

Return Type numeric

Default ON (1)

Last Modified: 22oct2017 New IQA command (A.10.1x)

[:SENSe]:SPEctrum:FFT:ANALysis:LENGth <num>

See also, If the SPEC:FFT:WIND and SPEC:FFT:LENG:AUTO.

(Read-Write) Set and query the FFT analysis length.

The “points” is the number of points for I/Q pairs. For example, if the Window Length is set to 10, it means the window length is for 10 I and 10 Q points. Not 5 I and 5 Q points.

See also SPEC:FFT:LENG.

Relevant Modes IQA

Parameters

<num> Minimum: 8
Maximum: 542288

Examples SPEC:FFT:ANAL:LENG 500

Query Syntax :SPEC:FFT:ANAL:LENG?

Return Type numeric

Last Modified: 22oct2017 New IQA command (A.10.1x)
[:SENSe]:SPECtrum:FFT:LENGth:AUTO <bool>

(Read-Write) Set and query the FFT analysis length control. This setting is directly coupled to the resolution bandwidth as follows: Enabling Length Control (AUTO) enables the automatic resolution bandwidth setting. Disabling automatic Length Control (AUTO) to MANual, disables automatic Resolution bandwidth (i.e., Analysis length and resolution BW can be set manually).

See also, If the SPEC:FFT:WIND and SPEC:FFT:ANAL:LENG.

**Relevant Modes** | IQA
---|---
**Parameters** | <boolean>

ON (1): Enables the automatic length control setting. The FFT analysis length control is set by the FieldFox. Resolution bandwidth is set to AUTO (i.e., the automatic resolution bandwidth by the FieldFox is enabled).

OFF (0): Disables the automatic length and FFT length settings. Analysis Length and resolution BW values can be set manually.

**Examples**

SPEC:FFT:LENG:AUTO 0

SPEC:FFT:LENG:AUTO ON

**Query Syntax** :SPEC:FFT:LENG:AUTO?

**Return Type** | numeric

**Default** | ON (1)

---

Last Modified: 22-Oct-2017 New IQA command (A.10.1x)

[:SENSe]:SPECtrum:FFT:WINDow <char>

See also, If the SPEC:FFT:LENG:AUTO and SPEC:FFT:ANAL:LENG.

(Read-Write) Set and query the FFT filtering Window setting (spectrum analysis window ).

**Relevant Modes** | IQA
---|---

**Parameters** | <character> UNIForm, TRIangular, HANNing, HAMMiing, BLACkman, BH4Tap, FLAT, FLATtop, FLAT6, FLAT7, P501, P601, KB70, KB90, KB110, GAUSsian60, GAUSsian70, GAUSsian80, SINC, TUKey25, BH3Tap, GUTop, KB80, KB100, KB120
Examples

SPEC:FFT:WIND HANN
SPEC:FFT:WIND KB70

Query Syntax

:SPEC:FFT:WIND?

Return Type

character

Default

FLATtop

Last Modified:

22-Oct-2017 New IQA command (A.10.1x)

[:SENSe]:SPECTrum:FREQuency:SPAN <freq>

(Read-Write) Set and query the frequency span in IQA (spectrum only). This is translated to the required Digital IF bandwidth for the FFT analysis. The analyzer’s Digital IF bandwidth is always equal to this value. The maximum span is equal to the maximum bandwidths of the IF Paths.

Relevant Modes

FOPS (Opt 208), IQA (Opt.351) - (Spectrum Only )

Parameters

<frequency> Span Frequency in Hz.
Minimum: 10 Hz
Maximum: 10 MHz

Couplings Changing the span causes the IFBW and SampleRate change automatically,
Span = IFBW, Sample Rate = 1.25 * IFBW
Res BW = Span /50, if Resolution Bandwidth = Auto.

Examples

SPEC:FREQ:SPAN 1e6

Query Syntax

SOURce:FREQuency:CENTer?

Return Type

Numeric

Default

10000000

Last modified:

22oct2017 New IQA mode (A.10.1x)

[:SENSe]:SWEep:ACQuisition <num>

(Read-Write) Set and query the sweep acquisition parameter. This effectively sets the sweep time in SA mode. Adjust this setting in order to increase the probability of intercepting and viewing pulsed RF signals.
Also set [:SENSe]:SWEep:ACQuisition:AUTO to 0 (OFF).
### Relevant Modes
SA, RTSA

### Parameters

#### <num>
Choose a relative acquisition value between 1 and 5000, where:
1 = Fastest sweep possible
5,000 = Slowest sweep possible.

### Examples
```plaintext
SWE:ACQ 25
```

### Query Syntax
`[:SENSe]:SWEep:ACQuisition?`

### Return Type
Numeric

### Default
1

---

Last Modified:
10-june-2016 Added RTSA mode (9.50)

### [:SENSe]:SWEep:ACQuisition:AUTO <bool>

(Read-Write) Set and query sweep acquisition state.

### Relevant Modes
SA, RTSA

### Parameters

#### <bool>
Choose from:
ON or 1 - Automatically set to the fastest sweep possible with the current settings.
OFF or 0 - Manually set the acquisition parameter using
`[:SENSe]:SWEep:ACQuisition`

### Examples
```plaintext
SWE:ACQ:AUTO 1
```

### Query Syntax
`[:SENSe]:SWEep:ACQuisition:AUTO?`

### Return Type
Boolean

### Default
ON or 1

---

Last Modified:
10-june-2016 Added RTSA mode (9.50)

### [:SENSe]:SWEep:MTIMe?

(Read-only) Query the measurement sweep time. This is the time reported on screen after a measurement completes.

Use with `INIT:IMM ; OPC?` in order to guarantee the most recently updated sweep time result.

### Relevant Modes
SA, NA, CAT, RTSA
Parameters  None

Examples  SWE:MTIM?

Return Type  Numeric

Default  Not Applicable

Last Modified:
10-june-2016  Added RTSA mode (9.50)
18-Mar-2014  New command (7.50)

[:SENSe]:SWEep:POINts <num>

(Read-Write) Set and query the number of data points in the trace.

Relevant Modes  CAT, LTE FDD, 5GTF, NA, NF, SA, RTSA, FOPS

VVM - Query only

Parameters

<num>  Number of data points. Choose a number from 2 to 10001.

Examples  SWE:POIN 250

Query Syntax  [:SENSe]:SWEep:POINts?

Return Type  Numeric

Default  401 - SA mode
201 - CAT, NA, FOPS
2 - VVM mode

Last Modified:
27nov2018  Added LTE FDD and 5GTF (11.0)
01-june-2018  Added Opt. 356 NF(10.3)
10-june-2016  Added RTSA mode (9.50)
20-Mar-2014  Added FOPS

[:SENSe]:SWEep:RX <value>

(Read-Write) Set and query the direction in which the receivers are swept.

Relevant Modes  FOPS

Parameters

<value>  Choose from the following:
FORward - Used for systems for which the output frequency is Offs + Src. The output frequency of the DUT and the receiver sweeps in the same direction as the source.

REVerse - Used for systems for which the output frequency is Offs - Src. The output frequency of the DUT and the receiver sweeps in reverse direction as compared to the source.

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SENSe]:SWEep:RX?</td>
<td>Character</td>
<td>FORWARD</td>
</tr>
</tbody>
</table>

Last modified:

19-Mar-2014 New command (A.07.50)

[:SENSe]:SWEep:TIME <num>

(Read-Write) Set and query the sweep time of the measurement. The actual sweep time that is displayed on the screen will usually be higher than this value due to the overhead sweep time.

In SA mode, use this command for Zerospan measurements.

To set and read sweep time for Non-zerospan measurements in SA mode, use [:SENSe]:SWEep:ACQuisition.

Relevant Modes CAT, NA, SA, RTSA

Parameters

<table>
<thead>
<tr>
<th>&lt;num&gt;</th>
<th>Sweep time in seconds.</th>
</tr>
</thead>
</table>

Examples SWE:TIME .250

Query Syntax [:SENSe]:SWEep:TIME?

Return Type Numeric

Default 0

Last Modified:

10-june-2016 Added RTSA mode

17-Aug-2012 Added SA mode

[:SENSe]:SWEep:TYPE <char>

(Read-Write) Set and query the sweep type.

Relevant Modes SA, RTSA
Parameters

<char> Choose from:

AUTO - FieldFox chooses the sweep type that yields the most accurate data most efficiently. When ResBW is set to 200 kHz and below, FFT is selected. Above 200 kHz, STEP is selected.

FFT - Use FFT sweep.

STEP - Use STEP sweep.

Examples

[SWEep:TYPE STEP]

Query Syntax

[:SENSe]:SWEep:TYPE?

Return Type

Character

Default

AUTO

Last Modified:

10-June-2016 Added RTSA mode (9.50)

1-Feb-2011 New command

[:SENSe]:SWEep:TYPE <value>

(Read-Write) Set and query the sweep type.

Relevant Modes

FOPS (Opt 208)

Parameters

<value> Choose from the following:

CW - Used to make standard (non-offset) power meter measurements at a single frequency.

SWEPt - Used to make swept FOPS measurements.

Examples

[SWEep:TYPESWEPt]

Query Syntax

[:SENSe]:SWEep:TYPE?

Return Type

Character

Default

CW

Last modified:

19-Mar-2014 New command (A.07.50)

[:SENSe]:TALlisten:AVOLume <num> - Obsolete
This command is replaced with `SYSTem:AUDio:VOLume`

(Read-Write) Set and query the Tune and Listen volume. To enable TuneListen mode, use `[:SENSe]:MEASurement:TAListen`

**Relevant Modes** SA

**Parameters**

<char> Tune and Listen volume. Choose a value in percent between 0 and 100 (loudest).

**Examples** `TAL:AVOL 50`

**Query Syntax** `[:SENSe]:TAListen:AVOLume?`

**Return Type** Numeric

**Default** 60

`[:SENSe]:TAListen:DSTate <bool>`

(Read-Write) Set and query the Tune and Listen demodulation state. To enable TuneListen mode, use `[:SENSe]:MEASurement:TAListen`

**Relevant Modes** SA

**Parameters**

<bool> Tune and Listen demod state. Choose from:

- OFF or 0 - Do not demodulate. Stops the audio demodulation and performs only the normal SA sweeps
- ON or 1 - Demodulate

**Examples** `TAL:DST 0`

**Query Syntax** `[:SENSe]:TAListen:DSTate?`

**Return Type** Boolean

**Default** ON

`[:SENSe]:TAListen:DTYPe <char>`

(Read-Write) Set and query the Tune and Listen demodulation type. This command allows you to change the demod type without enabling Tune and Listen. To enable TuneListen mode and set the demod type, use `[:SENSe]:MEASurement:TAListen`

**Relevant Modes** SA

**Parameters**

<char> Tune and Listen demodulation type. Choose from:

- AM - Amplitude Modulation.
- FMN - FM Narrow
- FMW - FM Wide
Examples TAL:DTYP AM
Query Syntax [:SENSe]:TAListen:DTYPe?
Return Type Character
Default AM

[:SENSe]:TAListen:LTIme <num>

(Read-Write) Set and query the Listen time for Tune & Listen.
While Tune & Listen is actively demodulating a signal, the SA does not sweep and update the display. Listen Time sets the amount of time that the FieldFox demodulates. It then stops to perform a single sweep and update the display, then again demodulates for this amount of time.
To enable TuneListen mode, use [:SENSe]:MEASurement:TAListen

Relevant Modes SA
Parameters
<char> Listen time in seconds. Choose a value between 0.1 and 100.
Examples TAL:LTIM 20
Query Syntax [:SENSe]:TAListen:LTIme?
Return Type Numeric
Default 2.5

[:SENSe]:TAListen:TFRq <num>

(Read-Write) Set and query the tune frequency for Tune & Listen. First set [:SENSe]:MEASurement:TAListen.
The Tune & Listen tuner is separate from the SA display. This allows you to listen to one frequency while displaying a different range of frequencies.
See Also: CALCulate:MARKer:SET
To enable TuneListen mode, use [:SENSe]:MEASurement:TAListen

Relevant Modes SA
Parameters
<char> Tune frequency in Hz. Choose a value between the FieldFox MIN and MAX frequency.
Examples TAL:TFR 101.7e6
Query Syntax [:SENSe]:TAListen:TFRq?
Return Type Numeric
Default 3.0 GHz
[:SENSe]:TOL <value>

(Read-Write) Set and query the power sensor measurement tolerance. Each power sensor reading is "settled" when either:

- two consecutive readings are within this Tolerance value or
- when the Max Number of Readings (:SENSe:POINT:READ:MAX) has been met.

The readings that were taken are averaged together to become the "settled" reading.

**Relevant Modes**  
FOPS (Opt 208)

**Parameters**

- <value>  
  Tolerance in dB. Choose a value between 0.0 and 10.0
  When consecutive power meter readings are within this value of each other, then the reading is considered settled.

**Examples**

| TOL .05 |

**Query Syntax**

[:SENSe]:TOL?

**Return Type**

Numeric

**Default**

.1

Last modified:

19-Mar-2014  
New command (A.07.50)

---

[:SENSe]:TRACe[:DATA]?

(Read-only) Read the data from a Power Meter measurement.

For Pulse Measurements ONLY: This command returns a single 'Meter-style' value. To read data for a trace graph, use CALCulate[:SELeected]:TRACe:DATA

**Relevant Modes**

- Power Meter, Pulse Measurements, CPM

**Examples**

| TRAC? |

**Query Syntax**

[:SENSe]:TRACe[:DATA]?

**Return Type**

Numeric

**Default**

Not Applicable

Last Modified:

1-Apr-2014  
Added CPM

29-Oct-2013  
Added Pulse

[:SENSe]:TRACe:LIMit:LOWer <num>
(Read-Write) Set and query the minimum (lower) limit value. Also set :SENSe:TRACe:LIMit:LOWer:STATe ON.

**Parameters**

<num>  Minimum (lower) limit value.

**Examples**  TRAC:LIM:LOW -50

**Query Syntax**  [:SENSe]:TRACe:LIMit:LOWer?

**Return Type**  Numeric

**Default**  -80

Last modified:

1-Apr-2014  Added CPM

31-Oct-2013  Added Pulse

[:SENSe]:TRACe:LIMit:LOWer:STATe <bool>

(Read-Write) Set and query the ON|OFF state for minimum (lower) limit testing.

**Parameters**

<bool>  Minimum limit state. Choose from:

0 or OFF - Minimum limit OFF

1 or ON - Minimum limit ON

**Examples**  TRAC:LIM:LOW:STATe 0

**Query Syntax**  [:SENSe]:TRACe:LIMit:LOWer:STATe?

**Return Type**  Boolean

**Default**  OFF

Last modified:

1-Apr-2014  Added CPM

31-Oct-2013  Added Pulse

[:SENSe]:TRACe:LIMit:UPPer <num>

(Read-Write) Set and query the maximum (upper) limit value. Also set :SENSe:TRACe:LIMit:UPPer:STATe ON.
Relevant Modes  
Power Meter, Pulse Measurements, CPM

Parameters

\(<\text{num}>\)  
Maximum (upper) limit value.

Examples  
\text{T}R\text{AC}:\text{LIM}:\text{UPP} \ 0

Query Syntax  
\[:\text{SE}\text{NSe}\]:\text{T}R\text{AC}:\text{LIM}:\text{UPP}\text{r}\text{?}\n
Return Type  
Numeric

Default  
-20

---

Last modified:

1-Apr-2014  Added CPM
31-Oct-2013  Added Pulse

\[:\text{SE}\text{NSe}\]:\text{T}R\text{AC}:\text{LIM}:\text{UPP}\text{r}:\text{STATe} <\text{bool}>

(Read-Write) Set and query the ON|Off state for maximum (upper) limit testing.

Relevant Modes  
Power Meter, Pulse Measurements, CPM

Parameters

\(<\text{bool}>\)  
Maximum limit state. Choose from:

0 or OFF - Maximum limit OFF
1 or ON  - Maximum limit ON

Examples  
\text{T}R\text{AC}:\text{LIM}:\text{UPP}:\text{STATe} \ 0

Query Syntax  
\[:\text{SE}\text{NSe}\]:\text{T}R\text{AC}:\text{LIM}:\text{UPP}\text{r}:\text{STATe}\text{?}\n
Return Type  
Boolean

Default  
OFF

---

Last modified:

1-Apr-2014  Added CPM
31-Oct-2013  Added Pulse

\[:\text{SE}\text{NSe}\]:\text{T}R\text{AC}:\text{MEAS}\text{urement}:\text{REF}erence <\text{num}>

(Read-Write) Set and query the Pulse Top setting which adjusts the measurement reference.

Relevant Modes  
Pulse Measurements

Parameters

\(<\text{num}>\)  
Pulse Top in percent. Choose a value between 0 and 100.
Examples

TRAC:MEAS:REF 90

Query Syntax

[:SENSe]:TRACe:MEASurement:REFerence?

Return Type

Numeric

Default

100

Last Modified:

29-Oct-2013

New command

[:SENSe]:TRACe:MEASurement <value>

(Read-Write) Set and query the measurement to display.

Relevant Modes

FOPS

Parameters

<value>

Choose from the following:

SPOWer - Not available until source data is stored into memory. Displays only the source power memory trace.

OUTPower - Displays the raw output power at the USB power sensor.

GAIN - Not available until source data is stored into memory. Displays the Output power / Source power (memory trace).

Examples

TRAC:MEAS GAIN

Query Syntax

[:SENSe]:TRACe:MEASurement?

Return Type

Character

Default

OUTPower

Last modified:

19-Mar-2014

New command (A.07.50)

[:SENSe]:V5G:CCARrier<n>:ENABle

(Read-Write) Set and query the component carrier center frequency.

Relevant Modes

5GTF

Parameters

<n>

Component carrier (CC) number. If unspecified, value is set to 1 (CC0).

n = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 (Up to eight component carriers can be measured at one time.)
<numeric>  Choices:
ON (1): Enable the target carrier.
OFF (0): Disable the target carrier.

Examples  
V5G:CCAR3:EN AB ON
V5G:CCAR5:EN AB 0

Query Syntax  
V5G:CCAR:CENT?

Return Type  
number

Default  ON|OFF|OFF|OFF|OFF|OFF|OFF|OFF

Last Modified:  
19-Dec-2018  New command

[:SENSe]:V5G:CCARrier<n>:FREQuency:CENTer
(Read-Write) Set and query the component carrier center frequency.

Parameters  

Relevant Modes  5GTF

Parameters  

<n>  Component carrier (CC) number. If unspecified, value is set to 1 (CC0).
    n = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 (Up to eight component carriers can be measured at one time.)

<numeric>  Choices:
Maximum (Hz): Dynamic. Varies with the FieldFox model.
Minimum: 0 Hz

Examples  
V5G:CCAR5:FREQ:CENT 28e9
V5G:CCAR3:FREQ:CENT 28200000000

Query Syntax  
V5G:CCAR:FREQ:CENT?

Return Type  
number

Default  100000000

Last Modified:  
19-Dec-2018  New command

[:SENSe]:V5G:SCORr:THReshold <num>
(Read-Write) Set and query the sync correlation threshold value (%).

### Relevant Modes
5GTF

#### Parameters

- `<number>`
  - Value of the sync correlation threshold %.
  - Minimum: 0
  - Maximum: 100

#### Examples

- V5G:SCOR:THR

#### Query Syntax

- V5G:SCOR:THR

#### Default

- 1%

---

Last Modified:
27 Nov 2018
Added 5GTF mode (11.0)

---

[:SENSe]:WAVeform:STARt <num>

(Read-Write) Sets the start time of the waveform measurement.

#### Relevant Modes

- IQA

#### Parameters

- `n/a`
- `<numeric>`
  - Minimum: 0s
  - Maximum: Dynamic. Varies with other setting couplings.

Note: The TimeDomainDisplayLength is derived by the following formula.

\[
\text{TimeDomainDisplayLength [pts]} = \text{SampleRate[pts/sec]} \times (\text{StopTime[sec]} - \text{StartTime[sec]}) + 1
\]

The maximum value of TimeDomainDisplayLength is set to 524288 by considering the memory constraint and cpu pressure.

When the change of Start Time makes TimeDomainDisplayLength exceed the limitation, the Stop Time Time will change to satisfy TimeDomainDisplayLength limitation.

The Maximum value is determined by current sample Rate and max capture samples.

Start time is adjusted to the nearest sample point based on sample period.

The max TimeDomainDisplayLength is 524288.

#### Examples

- :WAV:STAR 3e-6
- :WAV:STAR .003

#### Query Syntax

- :WAV:STAR?

#### Return Type

- Numeric

#### Default

- 0.00000E+00
[:SENSe]:WAVeform:STOP <num>

(Read-Write) Sets the stop time of the waveform measurement.

**Relevant Modes**  IQA

**Parameters**  n/a

- `<numeric>` Minimum: 0s
  Maximum: Dynamic. Varies with other setting couplings.

  **Note:** The TimeDomainDisplayLength is derived by the following formula.

  
  \[
  \text{TimeDomainDisplayLength} [\text{pts}] = \text{SampleRate}[\text{pts/sec}] \times \\
  (\text{StopTime}[\text{sec}] - \text{StartTime}[\text{sec}]) + 1
  \]

  The maximum value of TimeDomainDisplayLength is set to 524288 by considering the memory constraint and cpu pressure.

  When the change of Stop Time makes TimeDomainDisplayLength exceed the limitation, the Start Time Time will change to satisfy TimeDomainDisplayLength limitation.

  The Maximum value is determined by current sample Rate and max capture samples.

  Start time is adjusted to the nearest sample point based on sample period.

  The max TimeDomainDisplayLength is 524288.

**Examples**

- `:WAV:STOP 6e-6`
- `:WAV:STOP .003`

**Query Syntax**  :WAV:STOP?

**Return Type**  Numeric

**Default**  0.00000E+00

---

Last Modified:

22-Oct-2017  Added IQA mode (10.1x)

---

SOURce:ENOABle <bool>

(Read-Write) Set and query the Source Enable ON|OFF setting. Set the power level using SOURce:POWer.

**Relevant Modes**  Power Meter and SA
Parameters

<boolean> Choose from:
- OFF - Source OFF
- ON - Source ON

Examples 
SOUR:ENAB ON

Query Syntax 
SOURce:ENABle?

Return Type 
Boolean

Default 
OFF

Last Modified:
1-Nov-2013 New command

SOURce:ENABle <bool>

Note: This command was previously named [:SENSe]:ISOurce:ENABle. That syntax still work, but the new syntax should be used for new development.

(Read-Write) Set and query the Independent source state.

Relevant Modes SA

Parameters

<boolean> Independent source state. Choose from:
- ON (1) - Independent source ON
- OFF (0) - Independent source OFF

Examples SOUR:ENAB 1

Query Syntax SOURce:ENABle?

Return Type Boolean

Default OFF

Last Modified:
25-Mar-2014 Changed name
20-Oct-2010 New command (5.30)

SOURce:FREQuency:CENTer <value>

(Read-Write) Set and query the center frequency. Also send SOURce:FREQuency:SPAN

Relevant Modes FOPS (Opt 208)
Parameters

<value> Center Frequency in Hz. The frequency limits for the measurement are
determined by the limits of both the FieldFox and the power sensor.

- The LOW frequency is limited by the higher of either the FieldFox or
  the power sensor minimum frequencies.
- The HIGH frequency is limited by the lower of either the FieldFox or
  the power sensor maximum frequencies.

Examples

SOUR:FREQ:CENT 1e9

Query Syntax

SOURce:FREQuency:CENTer?

Return Type

Numeric

Default

50 MHz

---

Last modified:

19-Mar-2014 New command (A.07.50)

---

SOURce:FREQuency[:CW] <num>

**Note:** This command was previously named [:SENSe]:ISOurce:FREQuency[:CW]. That syntax will still work, but the new syntax should be used for new development.

(Read-Write) Set and query the Independent Source CW frequency.

Use SOURce:MODE to set Independent Source to CW mode.

Use SOURce:ENABLE to enable Independent Source.

**Relevant Modes** SA

**Parameters**

<num> CW frequency. Choose a value within the IQA and SA mode frequency range.

**Examples** SOUR:FREQ 1e9

**Query Syntax** SOURce:FREQuency[:CW]?

**Return Type** Numeric

**Default** IQA, SA mode center frequency

---

Last Modified:

25-Mar-2014 Name change

10-Oct-2010 New command (5.30)

---

SOURce:FREQuency:SPAN <value>
*(Read-Write)* Set and query the frequency span of the measurement. Also send SOURce:FREQuency:CENTer

**Relevant Modes**  **FOPS** (Opt 208)

**Parameters**

<value> Frequency span in Hz. The frequency limits for the measurement are determined by the limits of both the FieldFox and the power sensor.

- The LOW frequency is limited by the higher of either the FieldFox or the power sensor minimum frequencies.
- The HIGH frequency is limited by the lower of either the FieldFox or the power sensor maximum frequencies.

**Examples** SOUR:FREQ:SPAN .5e9

**Query Syntax** SOURce:FREQuency:SPAN?

**Return Type** Numeric

**Default** 10 MHz

---

Last modified:

19-Mar-2014 New command (A.07.50)

---

**SOURce:FREQuency:STARt <value>**

*(Read-Write)* Set and query the start frequency of the measurement. Also send SOURce:FREQuency:STOP

**Relevant Modes**  **FOPS** (Opt 208)

**Parameters**

<value> Start frequency in Hz. The frequency limits for the measurement are determined by the limits of both the FieldFox and the power sensor.

- The LOW frequency is limited by the higher of either the FieldFox or the power sensor minimum frequencies.
- The HIGH frequency is limited by the lower of either the FieldFox or the power sensor maximum frequencies.

**Examples** SOUR:FREQ:STAR  .5e9

**Query Syntax** SOURce:FREQuency:STARt?

**Return Type** Numeric

**Default** 45 MHz

---

Last modified:
### SOURce:FREQuency:STOP <value>

**Description:** Set and query the stop frequency of the measurement. Also send SOURce:FREQuency:STARt

**Relevant Modes:** FOPS (Opt 208)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;value&gt;</td>
<td>Stop frequency in Hz. The frequency limits for the measurement are determined by the limits of both the FieldFox and the power sensor.</td>
</tr>
<tr>
<td></td>
<td>- The LOW frequency is limited by the higher of either the FieldFox or the power sensor minimum frequencies.</td>
</tr>
<tr>
<td></td>
<td>- The HIGH frequency is limited by the lower of either the FieldFox or the power sensor maximum frequencies.</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURce:FREQuency:STOP 1.5e9</td>
</tr>
</tbody>
</table>

**Query Syntax**

```
SOURce:FR EQuency:STOP?
```

**Return Type**

Numeric

**Default**

55 MHz

---

**Note:**

**IMPORTANT!** This command syntax still works, but it is recommended that you use: SOUR:TRAC for new development.

This command was previously named [:SENSe]:ISOurce:MODE. That syntax will still work, but this new syntax should be used for new development.

**Usage:** (Read-Write) Set and query the Independent Source mode setting.

**Use** SOURce:ENABle to enable Independent Source.

**Relevant Modes**

<table>
<thead>
<tr>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
</tr>
</tbody>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;character&gt;</td>
<td>Independent source mode. Choose from:</td>
</tr>
<tr>
<td>CW</td>
<td>The internal source is set to CW. Use SOURce:FREQuency[:CW] to set the frequency.</td>
</tr>
<tr>
<td>TCW</td>
<td>The internal source is set to tracking CW.</td>
</tr>
<tr>
<td>SRTS</td>
<td>“Stimulus Response Tracking Sweep”. The internal source tracks with</td>
</tr>
</tbody>
</table>
the SA receiver. Use `SOURce:NORMalize` to normalize the trace.

<table>
<thead>
<tr>
<th>Examples</th>
<th>SOUR:MODE CW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>SOURce:MODE?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Character</td>
</tr>
<tr>
<td>Default</td>
<td>CW</td>
</tr>
</tbody>
</table>

Last Modified:

- 25-Mar-2014 Name change
- 10-Oct-2010 New command (5.30)

**SOURce:NORMalize <bool>**

*Note:* This command was previously named [:SENSe]:ISOurce:NORMalize. That syntax will still work, but the new syntax should be used for new development.

*(Read-Write)* Set and query the state of Independent source normalization.

Independent source must be ON (SOURce:ENABle) and in tracking mode SOURce:TRACking.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;bool&gt; Normalization state. Choose from: ON (1) - Normalization ON OFF (0) - Normalization OFF</td>
</tr>
<tr>
<td>Examples</td>
<td>SOUR:NORM 1</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>SOURce:NORMalization?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Last Modified:

- 25-Mar-2014 Name change
- 10-Oct-2010 New command (5.30)

**SOURce:POWer <num>**

*(Read-Write)* Set and query the source power level. This command sets SOURce:POWer:ALC[:MODE] to MANual.

- For N9912A and N9923A models: This command changes the source attenuation (dB) to effectively set the power level at the test port.
- All other models: This command directly changes the power level in dBm.
Relevant Modes  CAT, NA, VVM, Power Meter

Parameters  

<num> Source power/attenuator level.
  
  • N9912A: 0 to -31 dB in 1 dB steps
  • N9923A: 0 to -47 dB in .5 dB steps
  • All other models: Set power level from +3 to -45 dBm in .1 dB steps.

This command also supports sending MIN and MAX power levels as arguments:
  
  • For N9912A and N9923A, MAX is equivalent to the HIGH setting. See the User's Guide for your FieldFox model.
  • For all other models, MAX is equivalent to the maximum leveled setting (+3)

Examples  

SOUR:POW -10  
Source:power max

Query Syntax  :SOURce:POWer?

Return Type  Numeric

Default  0

Last modified:

1-Nov-2013  Added Power Meter
3-Apr-2013  Major edits (LH)
15-Aug-2012  Added link to ALC mode
17-Jul-2012  Added all other models (A.06.00)
1-Feb-2011  Fixed and reworded to emphasize negative power levels.

SOURce:POWer <num>  

Note: This command was previously named [:SENSe]:SOurce:POWer. That syntax will still work, but the new syntax should be used for new development.

(Read-Write) Set and query the Independent Source power level. This command automatically sets SOURce:POWer:MAXimum to OFF.

Use SOURce:ENABLE to enable Independent Source.

Relevant Modes  RTSA, SA  (All models EXCEPT N9912A - See [:SENSe]:ISOurce:POWer:ATTenuation for this model).

Parameters  

<num> Power level in dBm. Choose a value between -45 and 0.
This command will accept MIN and MAX as arguments.
Examples
SOUR:POW -20
source:power min

Query Syntax
SOURce:POWer?

Return Type
Numeric

Default
-15 dBm

Last Modified:
25-Mar-2014 Name change
10-Oct-2010 New command (6.00)

SOURce:POWer:ALC[:MODE] <char>

(Read-Write) Set and query the source power ALC (automatic leveling control) setting.
To set source power to a specific level, use SOURce:POWer. (Automatically sets SOURce:POWer:ALC MAN.)

Relevant Modes
CAT, NA, VVM

Parameters
<char> Source power. Choose from:
- **HIGH** - highest power level at each frequency (NOT flat)
- **LOW** - low power level (same as SOUR:POW MIN setting). For FieldFox models other than N9912A and N9923A, power is flat over the selected frequency range.
- **MAN** - Manual setting (Automatically set when SOURce:POWer is set.)

Examples
SOUR:POW:ALC HIGH

Query Syntax
SOURce:POWer:ALC[:MODE]?

Return Type
Character

Default
HIGH

Last Modified:
3-Apr-2013 Edited leveling settings
15-Aug-2012 New command

SOURce:POWer:ATTenuation <num>

Note: This command was previously named [:SENSe]:ISO reverence:POWer:ATTenuation. That syntax will still


469
work, but the new syntax should be used for new development.

(Read-Write) Set and query the Independent Source attenuation level.
Use SOUrce:ENABle to enable Independent Source.

Relevant Modes SA (N9912A ONLY - See SOUrce:POWer:MAXimum and SOUrce:POWer for all other models).

Parameters

<num> Independent source attenuation in dB. Choose a value between 31 and 0. This command will accept MIN and MAX as arguments.

Examples SOUR:POW:ATT 20
source:power:att min

Query Syntax SOURce:POWer:ATTenuation?

Return Type Numeric

Default 10

Last Modified:

7-Apr-2014 Name change
10-Oct-2010 New command (5.30)

SOUrce:POWer:MAXimum <bool>

Note: This command was previously named [:SENSe]:ISource:POWer:MAXimum. That syntax will still work, but the new syntax should be used for new development.

(Read-Write) Set and query the state of maximum Independent Source Power.
Use SOUrce:ENABle to enable Independent Source.

Relevant Modes SA (All models EXCEPT N9912A - See SOUrce:POWer:ATTenuation for this model).

Parameters

<bool> Max Independent Source Power state. Choose from:
ON (or 1) - Independent Source Power level is set to the maximum achievable power at all frequencies.
OFF (or 0) - Independent Source Power level is set using SOUrce:POWer.

Examples SOUR:POW:MAX 1
source:power:maximum off

Query Syntax SOURce:POWer:MAXimum?

Return Type Boolean

Default OFF
SOURce:POWer:MEMorize

(Write-Only) Stores the current data trace into memory.

Relevant Modes  FOPS

Examples  SOUR:POW:MEM

Query Syntax  Not Applicable

Default  Not Applicable

Last modified:

19-Mar-2014  New command

SOURce:POWer <num>

(Read-Write) Set and query the source power level. This command sets SOURce:POWer:ALC[:MODE] to MANual.

• For N9912A and N9923A models: This command changes the source attenuation (dB) to effectively set the power level at the test port.
• All other models: This command directly changes the power level in dBm.

Relevant Modes  CAT, NA, VVM, Power Meter

Parameters

<num>  Source power/attenuator level.

• N9912A: 0 to -31 dB in 1 dB steps
• N9923A: 0 to -47 dB in .5 dB steps
• All other models: Set power level from +3 to -45 dBm in .1 dB steps.

This command also supports sending MIN and MAX power levels as arguments:

• For N9912A and N9923A, MAX is equivalent to the HIGH setting. See the User's Guide for your FieldFox model.
• For all other models, MAX is equivalent to the maximum leveled setting (+3)

Examples  SOUR:POW -10

Source:power max

Query Syntax  :SOURce:POWer?
**SOURce:POWer <num>**

**Note:** This command was previously named [:SENSe]:SOurce:POWer. That syntax will still work, but the new syntax should be used for new development.

*(Read-Write)* Set and query the Independent Source power level. This command automatically sets SOURce:POWer:MAXimum to OFF.

Use SOURce:ENABle to enable Independent Source.

**Relevant Modes**
RTSA, SA (All models EXCEPT N9912A - See [:SENSe]:ISOurce:POWer:ATTenuation for this model).

**Parameters**

<num> Power level in dBm. Choose a value between -45 and 0.
This command will accept MIN and MAX as arguments.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOUR:POW -20</td>
</tr>
<tr>
<td>source:power min</td>
</tr>
</tbody>
</table>

**Query Syntax**

SOURce:POWer?

**Return Type** Numeric

**Default** -15 dBm

---

Last Modified:

- 25-Mar-2014 Name change
- 10-Oct-2010 New command (6.00)
### Relevant Modes
**FOPS** (Opt 208)

#### Parameters

**<value>** Offset frequency in Hz. (The frequency that the receiver is offset from the source.) This is typically the frequency of the LO that is used with the frequency converter. The frequency offset may be positive, negative, or zero.

#### Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOUR:REC:OFFS</td>
<td>Numeric</td>
<td>0 Hz</td>
</tr>
</tbody>
</table>

#### Query Syntax

SOURce:RECeiver:OFFSet?

#### Return Type

Numeric

#### Default

0 Hz

---

**Last modified:**

19-Mar-2014 New command (A.07.50)

---

### :STATus:OPERation:SA Mode:CONDition?

*(Read-Only)* Reads the status of a RecordPlayback session.

#### Relevant Modes

SA, RTSA

#### Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT:OPER:SAM:COND?</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

#### Return Type

**Numeric:**

- 0 - Record Playback session is NOT OPEN
- 64 - Record Playback session is OPEN

#### Default

Not Applicable

---

**Last Modified:**

10-June-2016 Added RTSA mode (9.50)

---

### :STATus:QUEstionable:FREQuency:CONDition?

*(Read-Only)* Reads the status of External Reference unlock errors.

#### Relevant Modes

ALL

#### Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT:QUES:FREQ:COND?</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

#### Return Type

**Numeric:**

- 0 - External reference NOT unlocked.
- 2 - External reference unlocked.

#### Default

Not Applicable

---
:STATus:QUEStionable:INTegrity:CONDition?

(Read-Only) Reads the status of ADC Overrange errors.

**Relevant Modes**  ALL - However, SA is the only mode that reports ADC overrange.

**Examples**  STAT:QUES:FREQ:COND?

**Return Type**  Numeric:

- 0 - ADC is NOT Overrange
- 2 - ADC Overrange occurs.

**Default**  Not Applicable

:STATus:QUEStionable:LIMit:CONDition?

(Read-Only) Reads the status of limit line failures.

**Relevant Modes**  ALL - Limit lines are allowed in CAT, NA, **NF**, SA

**Examples**  STAT:QUES:LIM:COND?

The following C# excerpt assumes 4 traces, and uses a "bitwise and" of the return value:

```csharp
for (int i = 1; i < 5; ++i)
{
    if (((returnValueFromScpiCommandAsInt & ((int)(Math.Pow(2, i)))) > 0)
    {
        tracePassFail[i] = "Fail";
    }
    else
    {
        tracePassFail[i] = "Pass";
    }
}
```

**Return Type**  Numeric:

- 0 - All limit lines pass
- 2 - Any limit on trace#1 failed
- 4 - Any limit on trace #2 failed
- 8 - Any limit on trace #3 failed
- 16 - Any limit on trace #4 failed

The returned values are added together. For example, 6 would mean traces #1 and #2 failed, 12 would mean #2 and #3 failed, 30 would mean all 4 traces failed and so forth.

**Default**  Not Applicable
SYSTem:AUDio:MUTe <bool>

(Read-Write) Set and query the system volume MUTE state.

**Relevant Modes** ALL Modes.

**Parameters**

- `<char>` Choose from:
  - **ON (or 1)** - Volume muted.
  - **OFF (or 0)** - Volume NOT muted.

**Examples**

SYST:AUD:MUTE 0

**Query Syntax** SYSTem:AUDio:MUTe?

**Return Type** Boolean

**Default** OFF (or 0)

Last Modified:

4-Aug-2011  New command

SYSTem:AUDio:VOLume <num>

(Read-Write) Set and query the system volume level.

**Relevant Modes** ALL Modes.

**Parameters**

- `<num>` Volume level. Choose a value between 0 (lowest volume) and 100 (highest volume).

**Examples**

SYST:AUD:VOL 93

**Query Syntax** SYSTem:AUDio:VOLume?

**Return Type** Numeric

**Default** 75

Last Modified:

4-Aug-2011  New command
SYSTem:BATtery?

(Read-Only) Reads whether or not a battery is present in the FieldFox.

**Relevant Modes**
ALL

**Examples**
SYST:BATT?

**Return Type**
Boolean
- Battery is present
- Battery is not present

**Default**
Not Applicable

Last Modified:
18-Oct-2012 New command

SYSTem:BATtery:ABSCharge?

(Read-Only) Reads the absolute charge on the battery in percent.

**Relevant Modes**
ALL

**Examples**
SYST:BATT:ABSC?

**Return Type**
Numeric

**Default**
Not Applicable

SYSTem:BATtery:ACURrent?

(Read-Only) Reads the average current flowing from the battery in amperes.

**Relevant Modes**
ALL

**Examples**
SYST:BATT:ACUR?

**Return Type**
Numeric

**Default**
Not Applicable

SYSTem:BATtery:ARTTe?

(Read-Only) Returns the minutes of run time remaining based on running average of current being used.

**Relevant Modes**
ALL

**Examples**
SYST:BATT:ARTT?

**Return Type**
Numeric

**Default**
Not Applicable
SYSTem:BATTery:CHEMistry?
(Read-Only) Reads the chemistry type of the battery.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:CHEM?</td>
</tr>
<tr>
<td>Return Type</td>
<td>String</td>
</tr>
<tr>
<td>Default</td>
<td>LION (Lithium Ion)</td>
</tr>
</tbody>
</table>

SYSTem:BATTery:CURRent?
(Read-Only) Reads amount of current being consumed when operating from internal battery. If battery is charging, indicates amount of charging current.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:CURR?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

SYSTem:BATTery:CYCLes?
(Read-Only) Reads the number of charge cycles battery has experienced. Charge cycle defined as ≥80% change in relative state of charge.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:CYCLes?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

SYSTem:BATTery:DATE?
(Read-Only) Reads the date of manufacture of the battery.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:DATE?</td>
</tr>
<tr>
<td>Return Type</td>
<td>String</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

SYSTem:BATTery:FCAPacity?
(Read-Only) Reads the capacity of a full battery in milli-amp Hours. Theoretically, how long a full battery of this type should continue to provide energy.
<table>
<thead>
<tr>
<th>Command</th>
<th>Relevant Modes</th>
<th>Examples</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Read-Only) Reads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reads the present accuracy of the battery gauge in percent. If the error exceeds 10%, you should recondition the battery.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTem:BATTery:MFGname?</td>
<td>ALL</td>
<td>SYST:BATT:MFGname?</td>
<td>String</td>
<td>Keysight</td>
</tr>
<tr>
<td>(Read-Only) Reads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reads the name of the manufacturer of the battery.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Read-Only) Reads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reads the remaining battery capacity in hours.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Read-Only) Reads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reads the current charge compared to actual full capacity in percent. This number lowers with age and number of battery cycles.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SYSTem:BATTery:RTTE?

**(Read-Only)** Reads the minutes of run time remaining based on amount of current being used now.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:RTTE?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

### SYSTem:BATTery:SAVer <bool>

**(Read-Write)** Set and query the battery saver state.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;bool&gt;</td>
</tr>
</tbody>
</table>
| <bool>         | Choose from:
| OFF or 0       | Battery saver OFF. This leaves the source ON between sweeps. |
| ON or 1        | Battery saver ON |
| Examples       | SYST:BATT:SAV OFF |
| Query Syntax   | SYSTem:BATTery:SAVer? |
| Return Type    | Boolean |
| Default        | ON |

### SYSTem:BATTery:SN?

**(Read-Only)** Reads the serial number of the battery.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:SN?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

### SYSTem:BATTery:STATus?

**(Read-Only)** Reads the use status of the FieldFox battery.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:BATT:STAT?</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SYSTem:BATTery:TEMPerature?</td>
<td>(Read-Only) Reads the current battery temperature in degrees Celsius.</td>
</tr>
<tr>
<td>SYSTem:BATTery:VENDor?</td>
<td>(Read-Only) Reads the vendor / distributor of the battery.</td>
</tr>
<tr>
<td>SYSTem:BATTery:VOLTage?</td>
<td>(Read-Only) Reads the current battery voltage in volts.</td>
</tr>
<tr>
<td>SYSTem:DATE &lt;year,mo,dy&gt;</td>
<td>(Read-Write) Set and query the system date.</td>
</tr>
</tbody>
</table>

**Return Type**
- **Character**
  - FULL - Battery is fully charged
  - CHAR - Battery is charging
  - NOB - No battery present
  - DISC - Battery is discharging; no AC Adaptor is present.

**Default**
- Not Applicable

Last Modified: 18-Oct-2012 New command
**Relevant Modes**  ALL

**Parameters**

<yr,mo,dy>  Year, month, and day.

**Examples**  SYST:DATE 2008,10,16

**Query Syntax**  SYSTem:DATE?

**Return Type**  Comma-separated numeric

**Default**  Not Applicable

---

**SYSTem:DCSupply?**

(Read-Only) Reads whether the DC Supply is connected to the FieldFox.

**Relevant Modes**  ALL

**Examples**  SYST:DCS?

**Return Type**  Boolean

1 - DC Supply is connected

0 - DC Supply is NOT connected

**Default**  Not Applicable

---

Last modified:

17-Jul-2012  New command (A.06.00)

---

**SYSTem:ERASe <value>**

(Write-Read) Immediately erases all user data from the FieldFox.

Warning: there is no confirmation message.

**Relevant Modes**  ALL

**Parameters**  None

<value>  (String) Choose “USERDATA” (case sensitive)

**Examples**  SYST:ERAS "USERDATA"

**Query Syntax**  SYSTem:ERASe? "USERDATA"

Returns the date of the last time that user data was erased.

**Default**  Not Applicable

---

Last modified:

28-Mar-2013  Added read
### SYSTem:ERRor[:NEXT]? **(Read-Only)**

Read the next error in the error queue. Learn [How to Query the Error Queue](#).

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYST:ERR?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric, String</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

### SYSTem:GPS:CNOise?

This query returns a string containing Carrier to Noise (C/No dBHz) data for each satellite currently in view of the FieldFox GPS.

**Note:** The C/No values returned are not calibrated at the factory or warranted for accuracy. The numbers retrieved are passed unprocessed directly from the GPS receiver in use.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>GPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>SYSTem:GPS:CNOise</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>SYSTem:GPS:CNOise?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Comma separated string in the form &quot;sat#,cno,sat#,cno,...&quot;. If GPS mode is not enabled or there are no satellites in view, the query returns &quot;0,0&quot;. If there are 10 satellites in view, this query generates 20 numbers in the SCPI return string.</td>
</tr>
<tr>
<td>Default</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### SYSTem:GPS:DATA?

***(Read-only)*** Returns current GPS data in the following form: `<latitude>,<longitude>,<elevation (m)>,<timestamp (UTC)>`

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Examples</td>
<td>SYST:GPS:DATA?</td>
</tr>
<tr>
<td>Returns</td>
<td>&quot;38 28.88657 N,122 42.66682 W,152,2010-06-28 23:35:38Z&quot;</td>
</tr>
<tr>
<td>Return Type</td>
<td>Comma-separated numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>
SYSTem:GPS:DATA:LAST?

(Read-only) Returns the data for the last successfully-locked GPS read in the form: 
<latitude>,<longitude>,<elevation (m)>,<timestamp (UTC)>,<seconds since last read>

Relevant Modes  ALL Modes.

Parameters  None

Examples  SYST:GPS:DATA:LAST?

'Returns

"38 28.88657 N,122 42.66682 W,152,2010-06-28 23:35:38Z,4"

Return Type  Comma-separated numeric

Default  Not Applicable

Last modified:

20-Jul-2010  New command (A.05.33)

SYSTem:GPS:DISPlay:COORdinate:FORMat <char>

(Read-Write) Set and query the format of the Latitude / Longitude coordinates.

Relevant Modes  ALL Modes.

Parameters

<char>  Choose from:

DMS - degrees, minutes, seconds

DMM - degrees, decimal minutes

Examples  SYST:GPS:DISP:COOR:FORM DMM

Query Syntax  SYSTem:GPS:DISPlay:COORdinate:FORMat?

Return Type  Character

Default  DMS

Last modified:

20-Jul-2010  New command (A.05.33)

SYSTem:GPS:DISPlay:DISTance:UNIT <char>
**Command Reference**

(Read-Write) Set and query the units of elevation.

**Relevant Modes** ALL Modes

**Parameters**

<char> Choose from:

- FEET
- METers

**Examples** SYST:GPS:DISP:DIST:UNIT MET

**Query Syntax** SYSTem:GPS:DISPlay:DISTance:UNIT?

**Return Type** Character

**Default** METers

---

**Last modified:**

20-Jul-2010 New command (A.05.33)

---

**SYSTem:GPS:DISPlay:STATe <bool>**

(Read-Write) Set and query the GPS status line display state.

**Relevant Modes** ALL Modes

**Parameters**

<char> Choose from:

- ON (1) - Status line ON
- OFF (0) - Status line OFF

**Examples** SYST:GPS:DISP:STAT 0

**Query Syntax** SYSTem:GPS:DISPlay:STATe?

**Return Type** Boolean

**Default** ON (1)

---

**Last modified:**

20-Jul-2010 New command (A.05.33)

---

**SYSTem:GPS:LSTate?**

(Read-only) Returns the GPS lock state.

**Relevant Modes** ALL Modes

**Parameters**

**Examples** SYST:GPS:LST?
**Return Type** Numeric
- 0 - OFF
- 1 - Locked
- 2 - ON but unlocked
- 3 - ON but no GPS device present

**Default**
0 - OFF

Last modified:
20-Jul-2010 New command (A.05.33)

**SYSTem:GPS[:STATe] <char>**

*(Read-Write)* Set and query the GPS state.

**Relevant Modes** ALL Modes.

**Parameters**

<char> Choose from:
- OFF - GPS OFF
- EXTERNAL - External GPS ON
- INTERNAL - Internal GPS ON. Requires Option 307 (Built-in GPS)

**Examples**

SYST:GPS EXT

**Query Syntax**

SYSTem:GPS[:STATe]?

**Return Type** Character

**Default**
OFF

Last modified:
17-Jul-2012 Updated for Internal (A.06.00)
20-Jul-2010 New command (A.05.33)

**SYSTem:GPS:SYNChronize <bool>**

*(Read-Write)* Set and query the GPS clock sync state.

**Relevant Modes** ALL Modes

**Parameters**

<char> Choose from:
- ON (1) - FieldFox clock synchronized with GPS clock.
- OFF (0) - FieldFox clock NOT synchronized with GPS clock.
SYSTem:PREFerences:DFLT

(Write-Only) Sets the system (language and display) preferences to their default settings.

Relevant Modes  ALL
Parameters  None
Examples  SYST : PREF : DFLT
Query Syntax  Not Applicable
Default  Not Applicable

SYSTem:PREFerences:SAVE

(Read-Write) Saves the current language and display settings as your system preferences.

Relevant Modes  ALL
Parameters  None
Examples  SYST : PREF : SAVE
Query Syntax  Not Applicable
Default  Not Applicable

:SYSTem:PRESet

(Write-Only) Perform a full preset on the FieldFox.

Relevant Modes  ALL
Examples  SYST : PRES
Query Syntax  Not Applicable
Default  Not Applicable

:SYSTem:PRESet:MODE

(Write-Only) Perform a Mode preset on the FieldFox. Only the current mode is preset to default settings.
**SYSTem:PWR:AUTO <value>**

*(Write-only)* Determines how the FieldFox is turned ON after a charged battery or DC power has been removed from the FieldFox. This setting will remain until it is changed either using the command or the User Interface.

**Relevant Modes**  ALL

**Parameters**

<value>

0 - Power comes ON only when the power button is pressed.
1 - Power comes ON automatically when either a charged battery or the DC adapter is inserted.

**Examples**

SYST:PWR:AUTO 1

**Query Syntax**  Not Applicable

**Default**  Not Applicable

---

**SYSTem:PWR:SHUTdown <value>**

*(Write-only)* Turns the FieldFox OFF. You can use with SYSTem:PWR:SHUTdown:DLY and SYSTem:PWR:SHUTdown:DURation.

**Relevant Modes**  ALL

**Parameters**

<value>

0 - Shuts down the FieldFox and does NOT reboot.
1 - Reboots the FieldFox

**Examples**

SYST:PWR:SHUT:DLY 5
SYST:PWR:SHUT:DUR 3
SYST:PWR:SHUT 0 //Shutdown after 5 seconds, no reboot
SYST:PWR:SHUT 1 //Reboot

**Query Syntax**  Not Applicable

**Default**  Not Applicable
SYSTem:PWR:SHUTdown:DLY <value>

(Read-Write) Sets the time to delay before turning the FieldFox OFF.
Use SYSTem:PWR:SHUTdown to turn the FieldFox OFF.
Use SYSTem:PWR:SHUTdown:DURation to specify period of time before rebooting.

Relevant Modes  ALL
Parameters
<value>  Time (in seconds) to delay shutdown.
Examples  Reboot after 5 seconds, for a duration of 3 seconds:
SYST:PWR:SHUT:DLY 5
SYST:PWR:SHUT:DUR 3
SYST:PWR:SHUT 1
Query Syntax  SYSTem:PWR:SHUTdown:DLY?
Default  0 - Do NOT delay

SYSTem:PWR:SHUTdown:DURation <value>

(Read-Write) Sets the time to wait before rebooting the FieldFox.
Use SYSTem:PWR:SHUTdown to turn the FieldFox OFF.

Relevant Modes  ALL
Parameters
<value>  Time (in seconds) to wait before rebooting the FieldFox.
Examples  Reboot after 5 seconds, for a duration of 3 seconds:
SYST:PWR:SHUT:DLY 5
SYST:PWR:SHUT:DUR 3
SYST:PWR:SHUT 1
Query Syntax  SYSTem:PWR:SHUTdown:DURation?
Default  -1 - Do NOT reboot after shutdown

SYSTem:PWR:SUSP <bool>

(Write-only) Puts the FieldFox into Standby mode. You can use with SYSTem:PWR:SUSP:DLY and SYSTem:PWR:SUSP:DURation.

Relevant Modes  ALL
Parameters
<bool>  Choose 1 to put the FieldFox into Standby.
IMPORTANT: There is no SCPI command to recover from standby mode.
Examples  SYST:PWR:SUSP 1
### SYSTem:PWR:SUSP:DLY <value>

**Query Syntax** Not Applicable  
**Default** Not Applicable

**SYSTem:PWR:SUSP:DLY <value>**  
(Read-Write) Sets the time to delay before putting the FieldFox in Standby.  
Use **SYSTem:PWR:SUSP** to put the FieldFox in Standby.  
Use **SYSTem:PWR:SUSP:DURation** to cause the FieldFox to awaken after a specified period of time.  

**Relevant Modes** ALL  
**Parameters**  
  - `<value>` Time (in seconds) to delay Standby.  
**Examples**  
<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTem:PWR:SUSP:DLY?</td>
<td>0 - Do NOT delay</td>
</tr>
</tbody>
</table>

### SYSTem:PWR:SUSP:DURation <value>

**Query Syntax** Not Applicable  
**Default** Not Applicable

**SYSTem:PWR:SUSP:DURation <value>**  
(Read-Write) Sets the time to wait before awakening the FieldFox from Standby.  
Use **SYSTem:PWR:SUSP** to put the FieldFox into Standby.  

**Relevant Modes** ALL  
**Parameters**  
  - `<value>` Time (in seconds) to wait before awakening the FieldFox.  
**Examples**  
<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTem:PWR:SUSP:DURation?</td>
<td>-1 Do NOT awaken after putting into standby</td>
</tr>
</tbody>
</table>

### SYSTem:UPReset:FPANel[:STATe] <bool>

**Query Syntax** Not Applicable  
**Default** Not Applicable

**SYSTem:UPReset:FPANel[:STATe] <bool>**  
(Read-Write) Set and query User Preset ON | OFF state.  

**Relevant Modes** ALL Modes.  
**Parameters**  
  - `<bool>` Choose from:  
    - OFF - User Preset OFF.  
    - ON - User Preset ON.  
**Examples**  
<table>
<thead>
<tr>
<th>Query Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTem:UPReset:FPANel[:STATe]?</td>
</tr>
</tbody>
</table>


SYSTem:UPReSet:MODE

*(Write-Only)* Perform a User MODE Preset on the FieldFox. The settings for only the current mode are recalled from UserPreset.sta.

**Relevant Modes** ALL

**Examples** SYST:UPR:MODE

**Query Syntax** Not Applicable

**Default** Not Applicable

SYSTem:UPReSet:SAVE

*(Write-Only)* Your FieldFox settings are saved to a standard State file (UserPreset.sta). However, unlike State files, calibration data is NOT saved.

**Relevant Modes** ALL

**Examples** SYST:UPR:SAVE

**Query Syntax** Not Applicable

**Default** Not Applicable

SYSTem:UPReSet

*(Write-Only)* Perform a User Preset on the FieldFox. The entire UserPreset.sta file is recalled. ALL modes assume the settings that were in place when the User Preset was saved. The active mode is the mode that was visible when the file was saved.

**Relevant Modes** ALL

**Examples** SYST:UPR

**Query Syntax** Not Applicable

**Default** Not Applicable

SYSTem:TIME <hr,min,sec>

*(Read-Write)* Set and query the current system time.

**Relevant Modes** ALL
Parameters
<num> Current time in hours (24 hr clock), minutes, seconds.

Examples
SYST:TIME 22,15,2

Query Syntax
SYSTem:TIME?

Return Type
Comma-separated numeric

Default
Not Applicable

SYSTem:TZONe <string>

(Read-Write) Set and query the current time zone setting.

Relevant Modes
ALL Modes.

Parameters
<string> Use SYSTem:TZONe:CATalog? to read the valid time zone settings.

Examples
SYST:TZON "(GMT+10:00) Brisbane"

Query Syntax
SYSTem:TZONe?

Return Type
String

Default
"(GMT-08:00) Pacific Time (US & Canada)"

Last modified:
20-Jul-2010 New command (A.05.30)

SYSTem:TZONe:CATalog?

(Read-only) Query the list of valid time zone settings. Use SYSTem:TZONe to set and read the current time zone setting.

Relevant Modes
ALL Modes.

Parameters
None

Examples
SYST:TZON:CAT?

Return Type
Comma-separated strings

Default
Not Applicable

Last modified:
20-Jul-2010 New command (A.05.30)

SYSTem:VERSion?
(Read-Write) Set and query the current version of the SCPI standard. For more information see: http://www.ivifoundation.org/docs/scpi-99.pdf

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Examples</td>
<td>SYST:VERS?</td>
</tr>
<tr>
<td>Return Type</td>
<td>String</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Last Modified:
16-Feb-2011 New command

SYSTem:VVS:CURRent?

(Read-only) Query the amount of current draw in amperes.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL Modes. ALL models with Option 309 (Voltage Source)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Examples</td>
<td>SYST:VVS:CURR?</td>
</tr>
<tr>
<td>Return Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Last Modified:
17-Jul-2012 New command

SYSTem:VVS:ENABle <bool>

(Write Only) Set and query the voltage source state.
To clear a 'Tripped' condition, send SYSTem:VVS:ENABle OFF, then SYSTem:VVS:ENABle ON.
To query the present state of the voltage source using SYSTem:VVS[:STATe]?

See example program.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>ALL Modes. ALL models with Option 309 (Voltage Source)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;bool&gt; Voltage source state. Choose from:</td>
</tr>
</tbody>
</table>

- ON (or 1) - Voltage source enabled.
- OFF (or 0) - Voltage source disabled.
SYST:VVS:ENAB 1

Query Syntax n/a
Return Type Boolean
Default OFF

Last Modified:
20june2018 Updated to be Write Only
23-Jul-2013 Added trip condition
17-Jul-2012 New command

SYSTem:VVS:HIMD <bool>

(Write Only) Set the high impedance load state.

Relevant Modes ALL Modes.
ALL models with Option 309 (Voltage Source)

Parameters

<bool> Voltage source state. Choose from:
    • ON (or 1) - High impedance load enabled.
    • OFF (or 0) - High impedance load disabled.

Examples SYST:VVS:HIMD 1
Query Syntax n/a
Return Type Boolean
Default OFF

Last Modified:
20june2018 New command

SYSTem:VVS:MAXCurrent?

(Read-only) Query the maximum amount of current that can be drawn at the present voltage setting.

Relevant Modes ALL Modes.
ALL models with Option 309 (Voltage Source)

Parameters None

Examples SYST:VVS:MAXC?
Return Type Numeric
### SYSTem:VVS:MVOLtage?

**(Read-only)** Query the present measured voltage.

**Relevant Modes**
ALL Modes.
ALL models with Option 309 (Voltage Source)

**Parameters**
None

**Examples**

```
SYST:VVS:MVOLtage?
```

**Return Type**
Numeric

**Default**
Not Applicable

---

**Last Modified:**
17-Jul-2012  New command

### SYSTem:VVS:RVOLtage <num>

**(Read Only)** Query the present requested voltage.

**Relevant Modes**
ALL Modes.
ALL models with Option 309 (Voltage Source)

**Parameters**
n/a

**Examples**

```
SYST:VVS:RVOL?
```

**Query Syntax**

```
SYSTem:VVS:RVOLtage?
```

**Return Type**
Numeric

**Default**
Varies by mode

---

**Last Modified:**
20June2018  New VVS command.

### SYSTem:VVS[:STATe]?
**SYSTem:VVS:VOLTage <num>**

(Read-Write) Set and query the output voltage.

To query the present requested voltage using `SYSTem:VVS:RVOLtage?`.

**Relevant Modes**
ALL Modes.
ALL models with Option 309 (Voltage Source)

**Parameters**

- `<num>` Output voltage. Choose from 1.00 to 32.00 with .01 V resolution.

**Examples**

- `SYST:VVS:VOLT 5.05`

**Query Syntax**

- `SYSTem:VVS:VOLTage?`

**Return Type**
Numerical

**Default**
1.00

---

**Last Modified:**

- 17-Jul-2012  New command
(Read-Only) Returns the current data trace values. If correction is ON, then the returned data is corrected.

- Set Units with [:SENSe]:AMPLitude:UNIT.
- Set data format with FORMat[:DATA]

Relevant Modes  **NF, SA, RTSA, VVM**
For **CAT mode** and **NA mode**, use **CALCulate:DATA:<type>** commands.

**Parameters**

<n>  Trace number of data to be returned.
SA Only. Choose from 1 through 4.
NF Only. Chose from 1 through 2.
If unspecified, <n> is set to 1.

<char>  **Note:** These parameters are NOT allowed for NF, RTSA and SA mode.
Type of data to return. Choose from:
**FDATA** - (VVM Only) Data in the current display format.
**SDATA** - (VVM Only) Raw (non-formatted) complex data.
VVM Mode always returns two data points: Mag and Phase.

**Examples**

```
TRACE:DATA? FDATA  'VVM mode. Assumes n = 1
TRAC2:DATA?  'NF, SA, & RTSA mode
```

**Return Type**  Comma-separated numeric
**Default**  Not Applicable

---

Last modified:

- 01june2018  Added NF mode Opt. 356 (10.3)
- 19-sep-2016  Added RTSA to relevant modes (A.09.50).
- 28-Oct-2011  Added link to format:data

**TRACe:ERTA:RINPut <char>**

(Read-Write) Set and query receiver trace measurement.

**Relevant Modes**  **ERTA**

**Parameters**

<string>  Choose from:
**B** - B receiver measurement.
**R** - R receiver measurement.
**BoR** - B over R (ratio) measurement.

**Examples**

```
TRACe:ERTA:RINPut BOR
```

**Query Syntax**  TRACe:ERTA:RINPut?
**Return Type**  Character

**Default**  BOR

Last Modified:

20-Jan-2015  New command (8.00)

**TRACe:IMAGinary:DATA?**

*(Read Only)* Returns the type the imaginary trace data (quadrature component vs. time) for the current trace.

**Note:** Only trace data is stored, **not** the screen's GUI data.

**Relevant Modes**  IQA

**Parameters**

<char>  n/a

**Examples**  TRAC: IMAG: DATA?

**Query Syntax**  TRAC:IMAG:DATA?

**Return Type**  real

**Default**  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).

**:TRACe:IMAGinary:Y:AUTO**

*(Write Only)* Enables the imaginary auto-scaling setting. The Auto Scaling function automatically determines the scale per division and reference values based on the measurement results.


**Relevant Modes**  IQA

**Parameters**  n/a

**Examples**  :TRAC: IMAG: Y:AUTO

**Query Syntax**  n/a

**Return Type**  n/a
**:TRACe:IMAGinary:Y:DLINe <volt>*

*(Read-Write) Set and query the waveform imaginary line values.*


**Note:** Polar trace is not supported.

**Relevant Modes**
- IQA

**Parameters**
- `<voltage>` Y-axis waveform imaginary level. Units depend on the selected setting. See `:TRAC:WAV:AMPL:UNIT`
  - Minimum: -9.9e37V
  - Maximum: 9.9e37V

**Examples**
- TRAC:IMAG:Y:DLIN 1

**Query Syntax**
`TRAC:IMAG:Y:DLINe?`

**Return Type**
Numeric

**Default**
0V

Last Modified:
- 22-oct-2017 Added IQA mode (10.1x)

---

**:TRACe:IMAGinary:Y:PDIVision <volt>*

*(Read-Write) Set and query the IQA waveform imaginary Y-axis scale values per division.*

**Relevant Modes**
- IQA

**Parameters**
- `<voltage>` Minimum: 1.00 nV
  - Maximum: 20.0V

**Examples**
- TRAC:IMAGinary:Y:PDIV 1.0e-9
- TRAC:IMAGinary:Y:PDIV 1
### TRACe:IMAGinary:Y:PDIV?

**Query Syntax** TRAC:IMAG:Y:PDIV?

**Return Type** numeric

**Default** 100 mV

Last Modified: 22-oct-2017 Added IQA mode (10.1x)

---

### :TRACe:IMAGinary:Y:RLEV <volt>

*(Read-Write)* Set and query the imaginary reference level value.


**Relevant Modes** IQA

**Parameters**

- `<voltage>` Y-axis waveform imaginary level.
  - Minimum: -5V
  - Maximum: 5V

**Examples** TRAC:IMAG:Y:RLEV 1

**Query Syntax** TRAC:IMAG:Y:RLEV?

**Return Type** Numeric

**Default** 0V

Last Modified: 22-oct-2017 Added IQA mode (10.1xx)

---

### :TRACe:IMAGinary:Y:RPOS <int>

*(Read-Write)* Set and query the imaginary reference position.


**Relevant Modes** IQA

**Parameters**

- `<integer>` Y-axis waveform imaginary reference position.
  - Minimum: 0
  - Maximum: 10
**Examples**

```plaintext
TRAC:IMAG:Y:RPOS 2
```

**Query Syntax**

```plaintext
TRAC:IMAG:Y:RPOS?
```

**Return Type**

Integer

**Default**

5

Last Modified:

22oct2017    Added IQA mode (10.1x)

---

**TRACe:IQCapture:DATA?**

*(Read Only)* Returns the type the I/Q capture data (quadrature component vs. time).

**Note:**
- This command can only be used in Single acquisition mode and after executing the `INITiate:IQCapture` command. Else, the data integrity cannot be guaranteed.

**Relevant Modes**

IQA

**Parameters**

n/a

**Examples**

```plaintext
TRAC:IQC:DATA?
```

**Query Syntax**

```plaintext
TRAC:IQC:DATA?
```

**Return Type**

numeric

**Default**

n/a

Last Modified:

19-Dec-2018    Added new IQA mode (10.1x).

---

**TRACe<n>:JITTer:WARNing:DATA?**

*(Read Only)* Reads the jitter goal exceeded trace data.

**Relevant Modes**

NF

**Parameters**

<n> Trace number.  
Choose from 1 to 2.

**Examples**

```plaintext
TRAC:JITT:WARN:DATA?  'Assumes n = 1
```

**Query Syntax**

```plaintext
TRACe2:JITTer:WARNIng:DATA?
```
**Return Type**  Block data

**Default**  1,1,1,1,1,1,1,1,1,1,1

---

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

**TRACe<n>:MEMory:DATA?**

*(Read Only)* Reads the normalized magnitude memory data for each of the trace points.

**Relevant Modes**  NF

**Parameters**

<n>  Trace number.
Choose from 1 to 2.

**Examples**  TRACe:MEM:DATA?  'Assumes n = 1

**Query Syntax**  TRACe2:MEMory:DATA?

**Default**  0.000000E+00,0.000000E+00

---

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

**TRACe<n>:MEMory:JITTer:WARNing:DATA?**

*(Read Only)* Reads the jitter goal exceeded warning memory trace value.

**Relevant Modes**  NF

**Parameters**

<n>  Trace number.
Choose from 1 to 2.

**Examples**  TRACe:MEM:JITT:WARN:DATA?  'Assumes n = 1

**Query Syntax**  TRACe2:MEMory:JITTer:WARNing:DATA?

**Default**  0,0

---

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)
**TRACe<n>:MEMory:UNCertainty:DATA?**

(Read-only) Read the uncertainty memory trace data (two values per data point -- upper uncertainty and lower uncertainty).


**Relevant Modes** NF

**Parameters**

<n> Trace number. Choose from 1 to 2.

**Examples**

TRACe2:MEM:UNC:DATA?  'Assumes n = 1

**Query Syntax** TRACe2:MEM:UNC:DATA?

**Return Type** Comma-separated numeric

**Default** Not Applicable

---

**Last Modified:**

01 June 2018 Added NF mode Opt. 356 (10.3)

---

**TRACe<n>:MEMory:UNCertainty:LOWer:DATA?**

(Read-only) Read the lower uncertainty memory trace data (one value per data point -- lower uncertainty).


**Relevant Modes** NF

**Parameters**

<n> Trace number. Choose from 1 to 2.

**Examples**

TRACe2:MEM:UNC:LOW:DATA?  'Assumes n = 1

TRACe2:MEM:UNC:LOW:DATA?

**Query Syntax** See Examples

**Return Type** Comma-separated numeric

**Default** Not Applicable

---

**Last Modified:**

01 June 2018 Added NF mode Opt. 356 (10.3)
TRACe<n>:MEMory:UNCertainty:UPPer:DATA?

(Read-only) Read the lower uncertainty memory trace data (one value per data point -- upper uncertainty).


Relevant Modes  NF

Parameters

<n> Trace number.
Choose from 1 to 2.

Examples  TRACe2:MEM:UNC:UPP:DATA?  'Assumes n = 1

Query Syntax  TRACe2:MEM:UNC:UPP:DATA?

Return Type  Comma-separated numeric

Default  Not Applicable

Last Modified:

01june2018  Added NF mode Opt. 356 (10.3)

TRACe:PHASe:DATA?

(Read Only) Returns the wrapped trace data (-180 degrees to 180 degrees).

Relevant Modes  IQA

Parameters

n/a

Examples  :TRACe:PHASe:DATA?

Query Syntax  :TRACe:PHASe:DATA?

Return Type  real

Default  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).

:TRACe:PHASe:Y:AUTO

(Read Only) Enables the phase auto-scaling setting. The Auto Scaling function automatically determines the scale per division and reference values based on the measurement results.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>Examples</td>
<td>:TRAC:PHAS:Y:_AUTO</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>n/a</td>
</tr>
<tr>
<td>Return Type</td>
<td>n/a</td>
</tr>
<tr>
<td>Default</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Last Modified:

22-oct-2017 Added IQA mode (10.1x)

:TRACe:PHASe:Y:DLINe <deg>

(Read-Write) Set and query the waveform phase view line values.


Note: Polar trace is not supported.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>TRAC:PHAS:Y:DLIN -180</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>TRAC:PHAS:Y:DLINe?</td>
</tr>
<tr>
<td>Return Type</td>
<td>degrees</td>
</tr>
<tr>
<td>Default</td>
<td>0 degrees</td>
</tr>
</tbody>
</table>

Last Modified:

22-oct-2017 Added IQA mode (10.1x)

:TRACe:PHASe:Y:RLEVel <deg>

(Read-Write) Set and query the phase reference level value.

Relevant Modes  IQA

Parameters

Y-axis waveform phase reference level.
Minimum:  -360 degrees
Maximum:  360 degrees

Examples  TRAC:PHAS:Y:RLEV  -180

Query Syntax  TRAC:PHAS:Y:RLEV?

Return Type  degrees

Default  0 degrees

Last Modified:
22oct2017  Added IQA mode (10.1x)

:TRACe:PHASe:Y:RPOSition <int>

(Read-Write) Set and query the phase reference position.

Relevant Modes  IQA

Parameters

Y-axis waveform phase reference position.
Minimum:  0
Maximum:  5

Examples  TRAC:PHAS:Y:RPOS  2

Query Syntax  TRAC:PHAS:Y:RPOS?

Return Type  Integer

Default  5

Last Modified:
22oct2017  Added IQA mode (10.1x)

TRACe:POLar:DATA?

(Read Only) Returns the polar trace data.
Relevant Modes | IQA
---|---
Parameters | n/a
Examples | :TRAC:POL:DATA?
Query Syntax | :TRAC:POL:DATA?
Return Type | real
Default | n/a

Last Modified:
22-oct-2017 Added new IQA mode (10.1x).

:TRACe:POLar:Y:AUTO

(Write Only) Enables the polar auto-scaling setting. The Auto Scaling function automatically determines the scale per division and reference values based on the measurement results.


Relevant Modes | IQA
---|---
Parameters | n/a
Examples | :TRAC:POL:Y:AUTO
Query Syntax | n/a
Return Type | n/a
Default | n/a

Last Modified:
22-oct-2017 Added IQA mode (10.1x)

:TRACe:POLar:Y:RLEVel <volt>

(Read-Write) Set and query the polar (imaginary vs. real) view reference level value.


Relevant Modes | IQA
Parameters

<voltage>  Y-axis waveform polar reference level.
Minimum: -5V
Maximum: 5V

Examples  TRAC:POL:Y:RLEV 0.02

Query Syntax  TRAC:POL:Y:RLEV?

Return Type  voltage

Default  500 mV

Last Modified:

22oct2017  Added IQA mode (10.1x)

TRACe:REAL:DATA?

(Read Only) Returns the real trace data (in phase component vs. time).

Relevant Modes  IQA

Parameters

n/a

Examples  TRAC:REAL:DATA?

Query Syntax  TRAC:REAL:DATA?

Return Type  real

Default  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).

TRACe:REAL:Y:AUTO

(Write Only) Enables the real auto-scaling setting. The Auto Scaling function automatically determines the scale per division and reference values based on the measurement results.

Relevant Modes  IQA
Parameters  n/a
Examples  

Query Syntax  n/a
Return Type  n/a
Default  n/a

Last Modified:
22-oct-2017  Added IQA mode (10.1x)

TRACe:REAL:Y:DLINe <volt>
(Read-Write) Set and query the waveform real line values.


Note: Polar trace is not supported.

Relevant Modes  IQA
Parameters
  <voltage>  Y-axis waveform real level. Units depend on the selected setting. See :TRAC:WAV:AMPL:UNIT
  Minimum: -9.9e37V
  Maximum: 9.9e37V

Examples  TRAC:REAL:Y:DLIN 1
Query Syntax  TRAC:REAL:Y:DLINe?
Return Type  Numeric
Default  0V

Last Modified:
22oct2017  Added IQA mode (10.1x)

:TRACe:REAL:Y:PDIVision <volt>
(Read-Write) Set and query the IQA waveform real Y-axis scale values per division.

Relevant Modes  IQA
Parameters

<voltage> Minimum: 1.00 nV
Maximum: 20.0V

Examples

TRAC:REAL:Y:PDIV 1.0e-9
TRAC:REAL:Y:PDIV 1

Query Syntax

TRAC:REAL:Y:PDIV?

Return Type

numeric

Default

100 mV

Last Modified:

22-oct-2017 Added IQA mode (10.1x)

TRACe:REAL:Y:RLEVel <volt>

(Read-Write) Set and query the waveform real reference level value.


Relevant Modes

IQA

Parameters

<voltage> Y-axis waveform real level.
Minimum: -5V
Maximum: 5V

Examples

TRAC:REAL:Y:RLEV 1

Query Syntax

TRAC:REAL:Y:RLEV?

Return Type

Numeric

Default

0V

Last Modified:

22oct2017 Added IQA mode (10.1x)

TRACe:REAL:Y:RPOSition <int>

(Read-Write) Set and query the real reference position.

Relevant Modes  IQA

Parameters

<integer>  Y-axis waveform real reference position.
Min: 0
Max: 10

Examples  TRAC:REAL:Y:RPOS 1

Query Syntax  TRAC:REAL:Y:RPOS?

Return Type  Integer

Default  5

Last Modified:
22oct2017  Added IQA mode (10.1x)

TRACe:SPECtrum:AMPLitude:SCALe <char>

(Read-Write) Set and query the IQA spectrum Y-axis scale to either logarithmic or linear.

Relevant Modes  IQA

Parameters

<char>  Scale type. Choose from:
LOG - units in dBm
LIN - units in dBm

Examples  TRAC:SPEC AMP:SCAL LIN

Query Syntax  TRAC:SPEC:AMPL:SCAL?

Return Type  Character

Default  LOG

Last Modified:
22-oct-2017  Added IQA mode (10.1x)

TRACe:SPECtrum:AMPLitude:UNIT <char>

(Read-Write) Set and query the IQA spectrum viewed line units, regardless of the current Scale setting. The UNIT choice affects the following: Reference Level, Trigger Level, Limit Lines, and Marker annotation. This measurement is FFT (Fast Fourier Transform) based.
Relevant Modes  IQA

Parameters

<character> The following are IQA (spectrum) only:

- DBM - dB milliWatts
- DBMV - dB milliVolts
- DBUV - dB microVolts
- DBMA - dB milliAmps
- DBUA - dB microAmps
- V - volts
- A - amps
- W - watts

Examples

TRAC:SPEC:AMPL:UNIT W
TRAC:SPEC:AMPL:UNIT DBM

Query Syntax

TRAC:SPEC:AMPL:UNIT?

Return Type  Character

Default  DBM

Last Modified:

22oct2017  Added IQA mode (10.1x)

TRACe:SPECtrum:RAWiq?

(Read only) Returns the spectrum raw I/Q data.

Relevant Modes  IQA

Parameters

n/a

Examples  :TRAC:SPEC:RAW?

Query Syntax  :TRAC:SPEC:RAW?

Return Type  real

Default  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).
:**TRACe:SPECTrum:Y:AUTO**

*(Write-Only)* Autoscale spectrum scale per division and the reference position values based on the measurement results.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>Examples</td>
<td>:TRAC:SPEC:Y:AUTO</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>N/A</td>
</tr>
<tr>
<td>Default</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Last Modified: 22oct2017 New IQA mode (10.1x)

:**TRACe:SPECTrum:Y:DLINe <ampl>**

*(Read-Write)* Set and query the spectrum display line Y-axis amplitude level.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;amplitude&gt;</td>
</tr>
<tr>
<td>Minimum</td>
<td>-9.9e37 dBm</td>
</tr>
<tr>
<td>Maximum</td>
<td>9.9e37 dBm</td>
</tr>
<tr>
<td>Examples</td>
<td>TRAC:SPEC:Y:DLIN 0</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>TRAC:SPEC:Y:DLIN?</td>
</tr>
<tr>
<td>Return Type</td>
<td>numeric</td>
</tr>
<tr>
<td>Default</td>
<td>-2.000E+01</td>
</tr>
</tbody>
</table>

Last modified: 22oct2017 New IQA mode (A.10.1x)

:**TRACe:SPECTrum:Y:PDIVision <num>**

*(Read-Write)* Set and query the IQA spectrum Y-axis scale values per division.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;numeric&gt;</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.0e-9 dB</td>
</tr>
<tr>
<td>Maximum</td>
<td>20 dB</td>
</tr>
</tbody>
</table>

Last modified: 22oct2017 New IQA mode (A.10.1x)
### :TRACe:SPECtrum:Y:RLEVel <amp>

*(Read-Write)* Set and query IQA spectrum Y axis absolute power reference value.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;numeric&gt; Enter IQA spectrum Y axis absolute power reference value.</td>
</tr>
<tr>
<td></td>
<td>Minimum: -210 dBm</td>
</tr>
<tr>
<td></td>
<td>Maximum: 30 dBm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th>:TRAC:SPEC:Y:RLEV 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>:TRAC:SPEC:Y:RLEV?</td>
</tr>
<tr>
<td>Return Type</td>
<td>numeric</td>
</tr>
<tr>
<td>Default</td>
<td>0.000000000E+00</td>
</tr>
</tbody>
</table>

Last Modified:
22-oct-2017 New IQA mode (10.1x)

### :TRACe:SPECtrum:Y:RPOStion <int>

*(Read-Write)* Set and query the position of the IQA spectrum Y-axis reference line. Use this command with TRAC:SPEC:Y:RLEVel which sets the Y-axis value of the absolute power reference value.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>&lt;integer&gt; Reference position. Choose a value from 0 to 10.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th>:TRAC:SPEC:Y:RPOS 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Syntax</td>
<td>:TRAC:SPEC:Y:RPOS?</td>
</tr>
<tr>
<td>Return Type</td>
<td>integer</td>
</tr>
</tbody>
</table>
**TRACe:SPECtrum<n>:DATA?**

*(Query only)* Returns spectrum trace data for the trace selected (1:4).

**Relevant Modes**  IQA

**Parameters**

<n>  Trace number for which display state is to be set or queried. Choose from 1 through 4.

<character>  Trace choice. Choose from: 1, 2, 3, or 4

**Examples**

TRAC:SPEC2:DATA?

**Query Syntax**

TRAC:SPEC<n>:DATA?

**Return Type**  real

**Default**  1

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).

---

**:TRACe:SPECtrum<n>:TYPE <char>**

*(Read-Write)* Set and query the spectrum trace state.

See also :TRAC:WAV<n>:TYPE.

**Relevant Modes**  IQA

**Parameters**

<n>  Trace number for which display state is to be set or queried. Choose from 1 through 4.

<char>  Trace type. Choose from:

- CLRW - Clear/Write
- BLANK - Blank
- MAXH - Max Hold
- MINH - Min Hold
AVG - Average (this parameter only applies to trace averaging.)

VIEW - View

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
</table>
| TRACe:SPEC<n>:TYPE VIEW | Character | CLWR - trace 1  
|  |  | BLAN - traces 2:4 |

Last Modified:

22oct2017 Added new IQA mode (10.1x).

TRACe:UPHase:DATA?

(Read Only) Returns the unwrapped trace data.

Relevant Modes

IQA

Parameters

n/a

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAC:UPH:DATA?</td>
<td>real</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Last Modified:

22-oct-2017 Added new IQA mode (10.1x).

:TRACe:UPHase:Y:AUTO

(Read Only) Enables the unwrapped phase auto-scaling setting. The Auto Scaling function automatically determines the scale per division and reference values based on the measurement results.


Relevant Modes

IQA

Parameters

n/a

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TRAC:UPH:Y:AUTO</td>
</tr>
</tbody>
</table>
**Query Syntax**  n/a  
**Return Type**  n/a  
**Default**  n/a

Last Modified:  
22-oct-2017  Added IQA mode (10.1x)

### :TRACe:UPHase:Y:DLINe

*(Read-Write)* Set and query the waveform unwrapped phase line values.


**Note:** Polar trace is not supported.

**Relevant Modes**  IQA

**Parameters**

- `<degree>`  Y-axis waveform unwrapped phase level. Units depend on the selected setting. See :TRAC:WAV:AMPL:UNIT
  - Minimum: -360 degrees
  - Maximum: 360 degrees

**Examples**  TRAC:UPH:Y:DLIN -180

**Query Syntax**  TRAC:UPH:Y:DLIN?

**Return Type**  degrees

**Default**  0 degrees

Last Modified:  
22-oct-2017  Added IQA mode (10.1x)

### :TRACe:UPHase:Y:RLEVel <deg>

*(Read-Write)* Set and query the unwrapped phase reference level value.


**Relevant Modes**  IQA

**Parameters**

- `<degree>`  Y-axis waveform unwrapped phase reference level.
Minimum: -360 degrees
Maximum: 360 degrees

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAC:UPH:Y:RLEV -180</td>
<td>degrees</td>
<td>0 degrees</td>
</tr>
</tbody>
</table>

Last Modified:
22oct2017  Added IQA mode (10.1x)

:TRACe:UPHase:Y:RPOSition <int>

(Read-Write) Set and query the unwrapped phase reference position.

Parameters

<integer> Y-axis waveform phase reference position.
Minimum: 0
Maximum: 5

Examples

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Return Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAC:UPH:Y:RPOS 2</td>
<td>Integer</td>
<td>5</td>
</tr>
</tbody>
</table>

Last Modified:
22oct2017  Added IQA mode (10.1x)

TRACe<n>:UNCertainty:DATA?

(Read-Only) Query the uncertainty trace data.
Note: Noise Figure has asymmetric uncertainty. Noise Factor and Noise Temperature, have symmetric uncertainty.

This query operates on the selected trace. First select a trace using CALC:PAR<n>:SELection.
See also, TRACe<n>:UNCertainty:DATA:UPPer? and TRACe<n>:UNCertainty:DATA:LOWer?.
Relevant Modes  NF
Parameters Not applicable - query only
Examples No write available
Query Syntax CALCulate:TRACe2:UNCertainty:DATA?
Return Type Block data
Default not applicable

Last Modified:
01 June 2018 Added NF mode Opt. 356 (10.3)

TRACe<n>:UNCertainty:LOWer:DATA?

(Read-Only) Query the uncertainty lower trace data.
Note: Noise Figure has asymmetric uncertainty. Noise Factor and Noise Temperature, have symmetric uncertainty.
This command operates on the selected trace. First select a trace using CALC:PAR<n>:SELect.
See also, TRACe<n>:UNCertainty:DATA? and TRACe<n>:UNCertainty:DATA:UPPer?.

Relevant Modes  NF
Parameters Not applicable - query only
Examples No write available
Query Syntax TRACe<n>:UNCertainty:DATA:LOWer?
Return Type Block data
Default not applicable

Last Modified:
01 June 2018 Added NF mode Opt. 356 (10.3)
(Read-Only) Query the uncertainty lower trace data.

**Note:** Noise Figure has asymmetric uncertainty. Noise Factor and Noise Temperature, have symmetric uncertainty.

This command operates on the selected trace. First select a trace using `CALC:PAR<n>:SELect`. See also, `TRACe<n>:UNCertainty:DATA?` and `TRACe<n>:UNCertainty:DATA:UPPer?`.

**Relevant Modes**  NF

**Parameters**

| Examples | No write available |

**Query Syntax**  TRACe<n>:UNCertainty:DATA:LOWer?

**Return Type**  Block data

**Default**  not applicable

---

Last Modified:

01june2018 Added NF mode Opt. 356 (10.3)

---

`:TRACe<n>:TYPE <char>`

(Read-Write) Set and query the type of SA Mode trace display state.

Use `[:SENSe]:QUANtity:TACTive?` to read the number of active traces.

**Relevant Modes**  SA, RTSA

**Parameters**

| <n> | Trace number for which display state is to be set or queried. Choose from 1 through 4. |
|<char> | Trace type. Choose from: |
|BRW | Clear/Write |
|BLANK | Blank |
|MAXH | Max Hold |
|MINH | Min Hold |
|AVG | Average (this parameter only applies to trace averaging.) |
|VIEW | View |

**Examples**  TRAC1:TYPE VIEW

**Query Syntax**  :TRACe<n>:TYPE?

**Return Type**  Character

**Default**  CLRW
Last Modified: 19-sep-2016 Added RTSA to relevant modes (A.09.50).

**TRACe:WAVeform:AMPLitude:SCALe <char>**

*(Read-Write)* Set and query the waveform RF envelope scale type.

**Relevant Modes** IQA

**Parameters**

*<character>* Scale type. Choose from:

- **LOG** - units in dBm
- **LIN** - units in mV

**Examples** TRAC:WAV:AMPL:SCAL LIN

**Query Syntax** TRAC:WAV:AMPL:SCAL?

**Return Type** Character

**Default** LOG

Last Modified: 22oct2017 Added IQA mode (10.1x)

**TRACe:WAVeform:AMPLitude:UNIT <char>**

*(Read-Write)* Sets and returns the output amplitude unit for trace of RF envelope vs. Time.

**Relevant Modes** IQA (RF envelope Only)

**Parameters**

*<character>* The following are IQA (spectrum) only:

- **DBM** - dB milliWatts
- **DBMV** - dB milliVolts
- **DBUV** - dB microVolts
- **DBMA** - dB milliAmps
- **DBUA** - dB microAmps
- **V** - volts
- **A** - amps
- **W** - watts

**Examples** TRAC:WAV:AMPL:UNIT W

TRAC:WAV:AMPL:UNIT DBM

**Query Syntax** TRAC:WAV:AMPL:UNIT?

**Return Type** Character
TRACe:WAVeform:RAWiq?

(Read Only) Returns the waveform raw I/Q trace data.

**Relevant Modes**  IQA

**Parameters**  n/a

**Examples**  :TRAC:WAV:RAW?

**Query Syntax**  :TRAC:WAV:RAW?

**Return Type**  real

**Default**  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).

:TRACe:WAVeform:Y:AUTO


**Relevant Modes**  IQA

**Parameters**  n/a

**Examples**  :TRAC:WAV:Y:AUTO

**Query Syntax**  n/a

**Return Type**  n/a

**Default**  n/a

Last Modified:

22-oct-2017  Added new IQA mode (10.1x).
:TRACe:WAVeform:Y:DLINe <ampl>

(Read-Write) Set and query the waveform RF envelope view line values.

See also TRACe:REAL:Y:DLINe, TRACe:IMAG:Y:DLIN, TRACe:PHAS:Y:DLIN, and TRACe:UPH:Y:DLIN.

Note: Polar trace is not supported.

Relevant Modes  IQA

Parameters

<amplitude>  Y-axis waveform amplitude level. Units depend on the selected setting. See :TRACe:WAVeform:AMPL:UNIT

Minimum: -9.9e37 dBm
Maximum: 9.9e37 dBm

Examples  TRACe:WAVeform:Y:DLIN 30

Query Syntax  TRACe:WAVeform:Y:DLIN?

Return Type  Numeric

Default  -20 dBm

Last Modified:

22-oct-2017  Added IQA mode (10.1x)

:TRACe:WAVeform:Y:PDIVision <rel_amp>

(Read-Write) Set and query the IQA RF envelope Y-axis scale values per division.

Relevant Modes  IQA

Parameters

<relative_amplitude>  Minimum: 1.0e-9 dB
Maximum: 20 dB

Examples  TRACe:WAVeform:Y:PDIVision 1.0e-9
TRACe:WAVeform:Y:PDIVision 5

Query Syntax  TRACe:WAVeform:Y:PDIVision?

Return Type  numeric

Default  1.000000000E+01
:TRACe:WAVeform:Y:RLEVel <amptd>

(Read-Write) Set and query IQA waveform RF envelope Y-axis absolute power reference level value.

- Relevant Modes: IQA
- Parameters:
  - <amplitude>: Enter IQA waveform Y-axis absolute power reference value.
    - Minimum: -210 dBm
    - Maximum: 90 dBm
- Examples: :TRAC:WAV:Y:RLEV 30
- Query Syntax: :TRAC:WAV:Y:RLEV?
- Return Type: numeric
- Default: 0.000000000E+00

Last Modified:
22-oct-2017 Added IQA mode (10.1x)

:TRACe:WAVeform:Y:RPOSirion <int>

(Read-Write) Set and query the position of the IQA waveform RF envelope Y-axis.

- Relevant Modes: IQA
- Parameters:
  - <integer>: Waveform reference position. Choose a value from 0 to 10.
- Examples: :TRAC:WAV:Y:RPOS 2
- Query Syntax: :TRAC:WAV:Y:RPOS?
- Return Type: Integer
- Default: 0
TRACe:WAVEform<n>:DATA?

(Read Only) Returns the RF envelope trace data (magnitude vs. power).

Relevant Modes  IQA

Parameters

<n> Trace number for which display state is to be set or queried. Choose from 1 through 4.

Examples  :TRACe:WAVE2:DATA?

Query Syntax :TRACe:WAVE<n>:DATA?

Return Type real

Default 1

Last Modified:
22-oct-2017  Added new IQA mode (10.1x).

:TRACe:WAVEform<n>:TYPE <char>

(Read-Write) Set and query the waveform RF envelope trace state.
See also :TRACe:SPEC<n>:TYPE.

Relevant Modes  IQA

Parameters

<n> Trace number for which display state is to be set or queried. Choose from 1 through 4.

<character> Trace type. Choose from:
CLRW - Clear/Write
BLANK - Blank
MAXH - Max Hold
MINH - Min Hold
AVG - Average (this parameter only applies to trace averaging.)
VIEW - View

Examples  :TRACe:WAVE2:TYPE MINH
          :TRACe:WAVE1:TYPE BLAN

Query Syntax :TRACe:WAVE<n>:TYPE?

Return Type Character
TRIGger:DELay <num>

(Write-Read) Set and query the trigger delay time.

Relevant Modes Pulse Measurements

Parameters

<num> Trigger delay time in seconds. Choose a value between -10 to +10.

Examples TRIG:DEL 5e-6

Query Syntax TRIGger:DELay?

Default 0

Last Modified:
29-Oct-2013 New command

TRIGger:LEVel:AUTO <bool>

(Write-Read) Set and query whether the trigger level is set manually or is set to the default level in the USB Power Sensor.

Relevant Modes Pulse Measurements

Parameters

<bool> Choose from:
ON or 1 - Trigger level is determined by the USB Power Sensor firmware.
OFF or 0 - Trigger level is set manually using TRIGger:LEVel.

Examples TRIG:LEV 5e-6

Query Syntax TRIGger:LEVel?

Default -15

Last Modified:
29-Oct-2013 New command
TRIGger:LEVel <num>

(Write-Read) Set and query the power level at which the USB power sensor is triggered when TRIGger:LEVel:AUTO = OFF.

Relevant Modes  Pulse Measurements

Parameters

<num> Trigger level in dBm.

Examples TRIG:LEV 5e-6

Query Syntax TRIGger:LEVel?

Default -15

Last Modified:
29-Oct-2013 New command

TRACe:PRESet:ALL

(Write-Only) Sets all of the RTSA traces back to their values.

Relevant Modes RTSA

Parameters

<char>

Examples TRACe:PRESet:ALL 'Both these examples reset all of the traces.
TRAC2:PRES:ALL

Return Type n/a

Default Not Applicable

Last modified:
19-sep-2016 Added new RTSA command (A.09.50).

:TRIGger[:SEQuence]:ATRigger <num>

(Read-Write) Set and query the auto-trigger time. If a trigger signal is not received before the specified
auto trigger time, a sweep will occur automatically.
Enable or disable automatic trigger time using :TRIG:ATR:STAT.

**Relevant Modes**  5GTF, IQA, SA, LTE FDD, RTSA

**Parameters**

- **<numeric>**  Auto trigger time in seconds.
  - SA: Choose a value between 0 and 65 seconds.
  - IQA and RTSA: Choose a value between 1 ms and 100 seconds.
  - Enter 0 (SA only) to set Auto Trigger OFF. When Auto Trigger is OFF, the FieldFox does NOT sweep unless a valid trigger signal is received.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:ATR 2</td>
</tr>
<tr>
<td>TRIGger:ATRigger .003</td>
</tr>
<tr>
<td>TRIG:ATR 1e2</td>
</tr>
</tbody>
</table>

**Query Syntax**  TRIG:ATR?

**Return Type**  Numeric

**Default**

- 1.000E+00 (SA)
- 1.00E-01 (RTSA)
- 1.000000E-01 (IQA)

---

**Last Modified:**

- 28-nov-2018  Added LTE FDD & 5GTF (11.0)
- 22-Oct-2017  Added IQA mode (10.1x)
- 10-June-2016  Added RTSA mode (9.50)
- 6-Mar-2013  Modified for 6.25

---

**:TRIGger[:SEQUence]:ATRigger:STATE <bool>**

*(Read-Write)*  Set and query the auto-triggering state.

Set automatic trigger time using :TRIG:ATR.

**Relevant Modes**  5GTF, IQA, SA, LTE FDD, RTSA

**Parameters**

- **<boolean>**  Auto trigger state. Choose from:
  - OFF or 0 - Auto Trigger OFF. The FieldFox does NOT sweep unless a valid trigger signal is received.
  - ON or 1 - Auto Trigger ON. If a trigger signal is not received before the specified Auto Trig Time (:TRIGger[:SEQUence]:EXTERNAL:ATRigger), a sweep will occur automatically.

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:ATR:STAT 1</td>
</tr>
</tbody>
</table>

**Query Syntax**  TRIGger[:SEQUence]:ATRigger:STATe?
### Return Type
- **Boolean**
- **Default**: OFF (0)

#### Last Modified:
- 28nov2018: Added LTE FDD & 5GTF (11.0)
- 22-Oct-2017: Added IQA mode (10.1x)
- 10-june-2016: Added RTSA mode (9.50)
- 6-Mar-2013: Modified for 6.25

#### :TRIGger[:SEQuence]:EXTernal:SLOPe <char>

**(Read-Write)** Set and query the trigger slope. Trigger Slope determines which edge of a trigger signal initiates a sweep.

**Note:** Only valid for Video, EXTERNAL, and RFBurst triggers.

**Relevant Modes**: 5GTF, IQA LTE FDD

**Parameters**
- `<character>`: Trigger slope. Choose from:
  - **POS**: Sweep is triggered by the rising (positive) edge of signal.
  - **NEG**: Sweep is triggered by the falling (negative) edge of signal.

**Examples**: `TRIG:EXT:SLOP NEG`

**Query Syntax**: `TRIGger[:SEQuence]:EXTernal:SLOPe?`

**Return Type**: Character
- **Default**: POS

#### Last Modified:
- 28nov2018: Added LTE FDD & 5GTF (11.0)
- 22-Oct-2017: Added new command for IQA mode (10.1x)

#### TRIGger[:SEQuence]:DELay <num>

**(Read-Write)** Set and query the trigger delay time. After a valid trigger signal is received, the sweep begins after the specified Trigger Delay time. Enable Trigger delay using `:TRIGger[:SEQuence]:EXTernal:DElay:STATe`

**Relevant Modes**: SA, RTSA

**Parameters**
- `<num>`: Trigger delay time in seconds. Choose a value between 0 and 65 seconds.
Examples

TRIG:DEL 2

Query Syntax
TRIGger[:SEQuence]:DELay?

Return Type
Numeric

Default
0

Last Modified:
10-june-2016 Added RTSA mode (9.50)
6-Mar-2013 Modified for 6.25

TRIGger[:SEQuence]:DELay:STATe

(Read-Write) Set and query the trigger delay state. Set delay time using TRIGger[:SEQuence]:DELay

Relevant Modes
SA, RTSA

Parameters
<bool>
Trigger delay state. Choose from:
OFF or 0 - Trigger delay OFF.
ON or 1 - Auto Trigger ON.

Examples
TRIG:DEL:STAT 1

Query Syntax
TRIGger[:SEQuence]:DELay:STATe?

Return Type
Boolean

Default
OFF (0)

Last Modified:
10-june-2016 Added RTSA mode (9.50)
6-Mar-2013 Modified for 6.25

:TRIGger[:SEQuence]:EXTernal:DELay <num>

(Read-Write) Set and query the trigger delay time. After a valid trigger signal is received, the sweep begins after the specified Trigger Delay time. Enable Trigger delay using :TRIGger[:SEQuence]:EXTernal:DELay:STATe

Relevant Modes
5GTF, IQA, LTE FDD

Parameters
<num>
Trigger delay time in seconds. Choose a value between 0 and 65 seconds.
IQA mode:
Minimum: -150 ms
Maximum: 500 ms

Examples

TRIG:EXT:DEL 2

Query Syntax
TRIGger[:SEQuence]:EXTernal:DELay?

Return Type
Numeric

Default
0

Last Modified:
28nov2018 Added LTE FDD & 5GTF (11.0)
22-Oct-2017 Added new IQA mode command (10.1x).

:TRIGger[:SEQuence]:EXTernal:DELay:STATe <bool>

(Read-Write) Set and query the trigger delay state. Set delay time using
:TRIGger[:SEQuence]:EXTernal:DELay

Relevant Modes
5GTF, IQA, LTE FDD

Parameters

<bool> Trigger delay state. Choose from:
OFF or 0 - Trigger delay OFF.
ON or 1 - Auto Trigger ON.

Examples
TRIG:EXT:DEL:STAT 1

Query Syntax
TRIGger[:SEQuence]:EXTernal:DELay:STATe?

Return Type
Boolean

Default
OFF (0)

Last Modified:
28nov2018 Added LTE FDD & 5GTF (11.0)
22-Oct-2017 Added new IQA mode command (10.1x).

:TRIGger[:SEQuence]:EXTernal:SLOPe <char>

(Read-Write) Set and query the trigger slope. Trigger Slope determines which edge of a trigger signal initiates a sweep.

Note: Only valid for VIDEO, EXTernal, and RFBurst triggers.

Relevant Modes
5GTF, IQA LTE FDD

Parameters

<character> Trigger slope. Choose from:
**Examples**

POS  - Sweep is triggered by the rising (positive) edge of signal.
NEG  - Sweep is triggered by the falling (negative) edge of signal.

**Query Syntax**

TRIGger[:SEQuence]:EXTernal:SLOPe?

**Return Type**

Character

**Default**

POS

---

**TRIGger[:SEQuence]:FGATe:DELay <num>**

*(Read-Write)* Set and query the trigger delay time. This effectively moves the gating area left (negative delay) or right (positive delay) within the gating window.

See all FFT Gating commands.

**Relevant Modes**

SA

**Parameters**

<num> Trigger delay time in seconds.

**Examples**

TRIG:DEL 2

**Query Syntax**

TRIGger[:SEQuence]:DELay?

**Return Type**

Numeric

**Default**

0

---

Last Modified:

28-Mar-2013  New command (6.25)

---

**TRIGger[:SEQuence]:FGATe[:STATe]**

*(Read-Write)* Set and query the FFT Gating On/Off state.

See all FFT Gating commands.

**Relevant Modes**

SA

**Parameters**

<bool> FFT Gating state. Choose from:

OFF or 0 - FFT Gating OFF.
ON or 1 - FFT Gating ON.
Examples | TRIG:FGAT 1
---|---
Query Syntax | TRIGger[:SEQUence]:FGATe[:STATe]?
Return Type | Boolean
Default | OFF (0)

Last Modified: 6-Mar-2013 Modified for 6.25

TRIGger[:SEQUence]:FGATe:VIEW[:STATe] <bool>

Not finished

(Read-Write) Set and query the display of the FT Gating window. When ON, a time domain window is displayed below the frequency domain window.

See all FFT Gating commands.

Relevant Modes | SA
Parameters | <num> Choose from:
| ON (or 1) - Gate window ON
| OFF (or 0) - Gate window OFF

Examples | TRIG:FGAT:VIEW 1
Query Syntax | TRIGger[:SEQUence]:FGATe:VIEW[:STATe]?
Return Type | Boolean
Default | OFF (or 0)

Last Modified: 28-Mar-2013 New command (6.25)

TRIGger[:SEQUence]:FGATe:VIEW:TIME <num>

Not finished

(Read-Write) Set and query the X-axis time span for the time domain window.

See all FFT Gating commands.

Relevant Modes | SA
Parameters | <num> FFT gating view time in seconds. Choose a value between seconds.

Examples | TRIG:FGAT:VIEW:TIME 2
Query Syntax | TRIGger[:SEQUence]:FGATe:VIEW:TIME?
Return Type  Numeric  
Default  1.5e-3  

Last Modified:  
28-Mar-2013  New command (6.25)  

TRIGger[:SEQUence]:FGATe:WIDTh <num>  
(Read-Write) Set and query the width of the gating area within the time domain window. The gating area is indicated by two vertical green lines.  
See all FFT Gating commands.  

Relevant Modes  SA  
Parameters  
<num>  Choose a value between 0 and 65 seconds.  

Examples  TRIG:FGAT:WIDT 2  

Query Syntax  TRIGger[:SEQUence]:FGATe:WIDTh?  
Return Type  Numeric  
Default  1e-3  

Last Modified:  
28-Mar-2013  New command (6.25)  

TRIGger[:SEQUence]:FRAMe:OFFSet  
(Read-Write) Set the trigger frame offset. This command lets you advance the phase of the frame trigger by the amount you specify. It does not change the period of the trigger waveform. If the command is sent multiple times, it advances the phase of the frame trigger an additional amount each time it is sent. See also period frame offset reset.  

Relevant Modes  RTSA  
Parameters  
<num>  Range: 0 to 100 ms  

Examples  TRIG:FRAM:OFFS 1.6 ms  

Query Syntax  [:SENSe]:FRAM:OFFS?  
Return Type  numeric  
Default  0.00000000
TRIGger[:SEQuence]:FRAMe:OFFSet:DISPlay:RESet

(Write-Only) Resets the trigger frame offset to 0. Resets the value of the periodic trigger frame offset display setting to 0.0 seconds. The current displayed trigger location may include an offset value defined with the Offset key. Pressing this key redefines the currently displayed trigger location as the new trigger point that is 0.0 s offset. The Frame Offset SCPI can then be used to add offset relative to this new timing.

**Relevant Modes**
- RTSA

**Parameters**
- <Bool>

**Examples**
- TRIG:FRAM:OFFS:DISP:RES

**Query Syntax**
- n/a

**Return Type**
- n/a

**Default**
- 0.00000000

Last Modified:
- 19-sep-2016 Added new RTSA command (A.09.50).

TRIGger[:SEQuence]:FRAMe:PERiod

(Read-Write) Sets the period of the internal periodic timer clock. For digital communications signals, this is usually set to the frame period of your current input signal.

**Relevant Modes**
- RTSA

**Parameters**
- <num>
  - Range: 100 ns (1.000E-07) to 400 ms (4.000000000E-1)

**Examples**
- TRIG:FRAM:PER 1.6 ms

**Query Syntax**
- TRIG:FRAM:OFFS?

**Return Type**
- numeric

**Default**
- 2.00000000E-02

Last Modified:
- 19-sep-2016 Added new RTSA command (A.09.50).
TRIGger[:SEQuence]:HOLDoff

(Read-Write) Sets the holdoff time between triggers. When the trigger condition is satisfied, the trigger occurs, the delay begins, and the holdoff time begins. New trigger conditions will be ignored until the holdoff time expires.

Relevant Modes  RTSA

Parameters

<num>  Range: 0 seconds (0.000000E+00) to 10 seconds (1.0000000E+01)

Examples  TRIG:HOLD 1.6 ms

Query Syntax  TRIG:HOLD?

Return Type  numeric

Default  0.000000E+00

Last Modified:

20-sep-2016  Added new RTSA command (A.09.50).

:TRIGger[:SEQuence]:LEVel

(Read-Write) Set and query the level at which a video or RF Burst trigger will occur. An incoming signal with this amplitude will initiate a sweep.

Relevant Modes  RTSA

Parameters

<num>  Trigger level. (Range: -210 dBm to 30 dBm)

Examples  TRIG:LEV -30

Query Syntax  TRIGger[:SEQuence]:LEVel?

Return Type  Numeric

Default  -25 dBm

Last Modified:

20-sep-2016  Added for RTSA mode A.09.50

TRIGger[:SEQuence]:POSition <num>

(Read-Write) Set and query the trigger position. Available ONLY in Zerospan measurements.

This is an easy way to automatically set the Trigger Delay by positioning the trigger event (also known as T zero) at any graticule along the X-axis.

Also set  TRIGger[:SEQuence]:POSition:STATe
**:TRIGger[:SEQUence]:POSition:STATe <bool>**

*(Read-Write)* Set and query the trigger position state. Set trigger position using TRIGger[:SEQUence]:POSition.

**Relevant Modes** SA

**Parameters**

<bool> Trigger position state. Choose from:
- **OFF** or 0 - Trigger position OFF.
- **ON** or 1 - Trigger position ON. TRIGger[:SEQUence]:DELay is set automatically and can NOT be overwritten.

**Examples** TRIG:POS:STAT 1

**Query Syntax** TRIGger[:SEQUence]:POSition:STATe?

**Return Type** Boolean

**Default** OFF (0)

---

Last Modified:

19-Dec-2018  New command
Relevant Modes | IQA
---|---
Parameters
<numeric> Minimum: -150 ms  
Maximum: 500 ms
Examples | TRIG:RFB:DEL 500e-3
Query Syntax | TRIG:RFB:DEL?
Return Type | Numeric
Default | 0.000000E+00 s

Last Modified:
22-Oct-2017 Added new IQA mode command (10.1x).

:TRIGger[:SEQUence]:RFBurst:DELay:STATe <bool>
(Read-Write) Set and query the IQA trigger delay states for RFBurst. Set delay time using :TRIG:RFB:DEL.

Relevant Modes | IQA
---|---
Parameters
<boolean> Trigger delay state. Choose from:
OFF or 0 - Trigger delay OFF.
ON or 1 - Auto Trigger ON.
Examples | TRIG:RFB:DEL:STAT 0
Query Syntax | TRIG:RFB:DEL:STAT?
Return Type | Boolean
Default | OFF (0)

Last Modified:
22-Oct-2017 Added new IQA mode command (10.1x).

:TRIGger[:SEQUence]:RFBurst:GLIMask <real>
(Read-Write) Set and query the IQA RF Burst glitch mask trigger to stabilize trigger signals and minimize
spurs.

**Relevant Modes**  
IQA

**Parameters**

<real>  
Minimum: 0 s  
Maximum: 5 ms

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:RFB:GLIM 3e-3</td>
<td></td>
</tr>
<tr>
<td>TRIG:RFB:GLIM .002</td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax**

TRIG:RFB:GLIM?

**Return Type**

Numeric

**Default**  
5.0000E-03

---

Last Modified:

22-Oct-2017  
Added new IQA mode command (10.1x).

---

**:TRIGger[:SEQuence]:RFBurst:LEVel**

(Read-Write) Set and query the IQA trigger level for RF Burst. Similar to a Video trigger where an acquisition is triggered from a signal at the SA RF Input connector. But, an RF Burst trigger is detected in the third IF stage. An acquisition is triggered when the amplitude of the incoming signal exceeds the settable Trigger Level.

**Note:** Only valid for RF Burst triggers.

**Relevant Modes**  
IQA

**Parameters**

<numerical>  
RF Burst trigger level (dBm).  
Minimum: -210 dBm  
Maximum: 30 dBm

**Examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:RFB:LEV -10</td>
<td></td>
</tr>
</tbody>
</table>

**Query Syntax**

:TRIG:VID:LEV?

:TRIG:RFB:LEV?

**Return Type**

numeric

**Default**  
-25 dBm

---

Last Modified:

22-Oct-2017  
New IQA command (A.10.1x)
TRIGger[:SEQuence]:RF Burst: SLOPe

(Read-Write) Set and query the IQA trigger slope. Trigger Slope determines which edge of a trigger signal initiates an acquisition.

Note: Only valid for VIDeo, EXTernal, and RF Burst triggers.

**Relevant Modes**: IQA

**Parameters**

<character> Trigger slope. Choose from:

- **POS**: Sweep is triggered by the rising (positive) edge of signal.
- **NEG**: Sweep is triggered by the falling (negative) edge of signal.

**Examples**: TRIG:RF Burst:SLOP NEG

**Query Syntax**: TRIGger:RF Burst:SLOPe?

**Return Type**: Character

**Default**: POS

---

Last Modified:

22-Oct-2017 Added new command for IQA mode (10.1x)

---

TRIGger[:SEQuence]: SLOPe

(Read-Write) Set and query the trigger slope. Trigger Slope determines which edge of a trigger signal initiates a sweep.

**Relevant Modes**: SA, RTSA

**Parameters**

<character> Trigger slope. Choose from:

- **POS**: Sweep is triggered by the rising (positive) edge of signal.
- **NEG**: Sweep is triggered by the falling (negative) edge of signal.

**Examples**: TRIG:SLOP NEG

**Query Syntax**: TRIGger[:SEQuence]:SLOPe?

**Return Type**: Character

**Default**: POS

---

Last Modified:

10-June-2016 Added RTSA mode (9.50)

28-Mar-2013 Modified for 6.25
**TRIGger[:SEQuence]:SOURce <char>**

(Read-Write) Set and query the source of FieldFox trigger signals.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA, 5GTF, SA, LTE FDD, RTSA</th>
</tr>
</thead>
</table>

**Parameters**

- `<character>`
  - Trigger slope. Choose from:
    - **FREE** - (Free run) Triggering is provided by the FieldFox internal circuitry. A new sweep begins when the previous sweep ends.
    - **EXT** - (External) A sweep is triggered on an external TTL signal at the External Trigger connector. External trigger is available only after first setting BNC Connector Use to Trigger.
    - **VID** - (Video) - *(Not applicable for LTE FDD/5GTF)* - Available in Zerospan and FFT sweeps. A sweep is triggered on a signal at the SA RF Input connector when the amplitude of the incoming signal exceeds the settable Trigger Level.
    - **RFB** - (RF Burst) - *(Not applicable for LTE FDD/5GTF)* - Available in Zerospan and FFT sweeps (SA and RTSA only). Similar to Video triggering, a sweep is triggered from a signal at the SA RF Input connector. However, an RF Burst trigger is detected in the third IF stage. A sweep is triggered when the amplitude of the incoming signal exceeds the settable Trigger Level.
    - **PER** - (Periodic) - *RTSA Only* - Trigger repeats at the period rate entered. Periodic trigger’s phase can be altered by the Offset value.

**See Also**

- TRIGger[:SEQuence]:VIDeo:LEVel (SA Only)
- TRIGger[:SEQuence]:VIDeo|RFBurst:LEVel (IQA Only)

**Examples**

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>TRIGger[:SEQuence]:SOURce?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Type</td>
<td>Character</td>
</tr>
<tr>
<td>Default</td>
<td>FREE</td>
</tr>
</tbody>
</table>

**Last Modified:**

- 28nov2018    Added LTE FDD & 5GTF (11.0)
- 22-Oct-2017  Added IQA mode (10.1x)
- 10-june-2016 Added RTSA mode (9.50)
- 28-Mar-2013  Added RF Burst (6.25)
**:TRIGger[:SEQuence]:VIDeo:DELay <num>**

*(Read-Write)* Set and query the IQA delay for video, external, or RF Burst triggers. An incoming signal with this amplitude will initiate an acquisition.

See also **TRIG:VID:DEL:STAT**.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>Minimum: -150 ms</td>
</tr>
<tr>
<td></td>
<td>Maximum: 500 ms</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>TRIG:VID:DEL 200e-3</td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td>TRIG:VID:DEL?</td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>Numeric</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>0.0000000E+00 s</td>
</tr>
</tbody>
</table>

Last Modified:
22-Oct-2017 Added new IQA mode command (10.1x).

**:TRIGger[:SEQuence]:VIDeo:DELay:STATe <bool>**

*(Read-Write)* Set and query the IQA trigger delay states for VIDeo. Set delay time using **:TRIG:VID:DEL**.

<table>
<thead>
<tr>
<th>Relevant Modes</th>
<th>IQA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>&lt;boolean&gt;</td>
<td>Trigger delay state. Choose from:</td>
</tr>
<tr>
<td></td>
<td>OFF or 0 - Trigger delay OFF.</td>
</tr>
<tr>
<td></td>
<td>ON or 1 - Auto Trigger ON.</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>TRIG:VID:DEL:STAT 1</td>
</tr>
<tr>
<td></td>
<td>TRIG:VID:DEL:STAT ON</td>
</tr>
<tr>
<td></td>
<td>TRIG:VID:DEL:STAT 0</td>
</tr>
<tr>
<td><strong>Query Syntax</strong></td>
<td>TRIG:VID:DEL:STAT?</td>
</tr>
<tr>
<td><strong>Return Type</strong></td>
<td>Boolean</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>OFF (0)</td>
</tr>
</tbody>
</table>

Last Modified:
22-Oct-2017 Added new IQA mode command (10.1x).
:TRIGger[:SEQUence]:VIDeo:LEVel <num>

(Read-Write) Set and query the level at which a video or RF Burst trigger will occur. An incoming signal with this amplitude will initiate a sweep.

**Relevant Modes** SA

**Parameters**

<num> Trigger level. The Units depend on the Scale Type setting: (dB for Log and mV for Linear). Set with [:SENSe]:AMPLitude:SCALe.

**Examples** TRIG:VID:LEV -30

**Query Syntax** TRIGger[:SEQUence]:VIDeo:LEVel?

**Return Type** Numeric

**Default** - 20 dBm (Log)

22.361 mv (Lin)

---

Last Modified:

28-Mar-2013 Modified for 6.25

---

:TRIGger[:SEQUence]:VIDeo:SLOPe <char>

(Read-Write) Set and query the IQA trigger slope. Trigger Slope for video determines which edge of a trigger signal initiates a acquisition.

**Note:** Only valid for VIDeo, EXTernal, and RF Burst triggers.

**Relevant Modes** IQA

**Parameters**

<char> Trigger slope. Choose from:

POS - Sweep is triggered by the rising (positive) edge of signal.
NEG - Sweep is triggered by the falling (negative) edge of signal.

**Examples** TRIG:VID:SLOP NEG

**Query Syntax** TRIGger:VIDeo:SLOPe?

**Return Type** Character

**Default** POS

---

Last Modified:

22-Oct-2017 Added new command for IQA mode (10.1x)
TRIGger:SLOPe <char>

(Read-Write) Set and query the polarity of a valid external trigger signal.

**Relevant Modes**  NA, Pulse Measurements

**Parameters**

<char> Choose from:

- **POSitive** - Sweep is triggered by the rising (positive) edge of signal at about 1.7 V.
- **NEGative** - Sweep is triggered by the falling (negative) edge of signal at about 1.0 V.

**Examples**

TRIG:SLOP POS

**Query Syntax**

TRIGger:SLOPe?

**Return Type**  Character

**Default**  POSitive

---

Last Modified:

31-Oct-2013  Added Pulse
6-Mar-2013  New command (6.25)

---

TRIGger:SOURce <char>

(Read-Write) Set and query the source of trigger signals which initiate a measurement.

See Trigger:Source command for Pulsed Measurements Mode.

**Relevant Modes**  NA

**Parameters**

<char> Choose from:

- **INTernal** - Sweeps are initiated by the FieldFox internal circuitry.
- **EXTernal** - A sweep is initiated on the rising or falling edge of an external TTL signal at the Ref In/Trig In connector on the FieldFox top panel.

**Examples**

TRIG:SOUR EXT

**Query Syntax**

TRIGger:SOURce?

**Return Type**  Character

**Default**  INTernal

---

Last Modified:

6-Mar-2013  New command (6.25)
UNIT:POWer <string>

(Read-Write) Set and query the units to display in Power Meter mode.

Relevant Modes  Power Meter
Parameters
<string>  Choose from:
"W"  (Watts)
"DBM"
Examples  UNIT:POW "W"
Query Syntax  UNIT:POWer?
Return Type  String
Default  DBM

:V5G:DATA:GPS?

(Read-Only) Returns 5GTF result with GPS information.

Relevant Modes  5GTF
Parameters  None
Examples  V5G:DATA:GPS?
Default  Not Applicable

Last Modified:
19-Dec-2018  New command

:V5G:DATA?

(Read-Only) Returns 5GTF result without GPS information.

Relevant Modes  5GTF
Parameters  None
Examples  V5G:DATA?
Default  Not Applicable

Last Modified:
19-Dec-2018  New command