

Keysight X-Series Signal Analyzers

This manual provides documentation for the following analyzers:

PXA Signal Analyzer N9030A

MXA Signal Analyzer N9020A

EXA Signal Analyzer N9010A

CXA Signal Analyzer N9000A

(SA Mode Only)

Notice: This document contains references to Agilent. Please note that Agilent's Test and Measurement business has become Keysight Technologies. For more information, go to www.keysight.com

PSA Programming
Compatibility
Guide

Notices

© Keysight Technologies 2008-2014

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

Trademark Acknowledgements

WiMAX, Mobile WiMAX, WiMAX Forum, the WiMAX Forum logo, WiMAX Forum Certified, and the WiMAX Forum Certified logo are US trademarks of the WiMAX Forum.

Manual Part Number

N9020-90192

Print Date

August 2014

Supersedes: August 2012

Printed in USA

Keysight Technologies
1400 Fountaingrove Parkway
Santa Rosa, CA 95403

Warranty

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

Technology Licenses

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

Restricted Rights Legend

If software is for use in the performance of a US Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as

“Restricted computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Keysight Technologies’ standard commercial license terms, and non-DOD Departments and Agencies of the US Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

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

WARNING

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

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/pxa>

<http://www.keysight.com/find/mxa>

<http://www.keysight.com/find/exa>

<http://www.keysight.com/find/cxa>

<http://www.keysight.com/find/mxe>

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

<http://www.keysight.com/find/emailupdates>

Information on preventing instrument damage can be found at:

<http://www.keysight.com/find/tips>

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/techsupport>

Table of Contents

List of Commands 11

1 Using This Guide

Command Compatibility Status Icons	25
Organization of this Manual	25
Measurement-Specific Commands	26

2 General Issues

Numeric Data Formatting	27
Sensitivity to Input Power Level	27
Band Break Location	27

3 Swept SA Measurement Setup

Switching to Spectrum Analyzer Measurement	29
ADC Dither	30
ADC Range & FFT IF Gain Auto	31
Average Count & Average /Hold Number	32
Average State	33
Average Type	34
Detector 2 & 3 State	35
Fixed / Relative Limit	36
Limit Enabling	37
Limit Line Data	38
Merge Limit Line Data	39
Phase Noise Optimization	40

4 PowerSuite Measurements Setup

General	42
Data Format for CONFigure Query	42
Measurement-Specific Commands	42
Example	42
Measurement Switching & Trigger Mode	42
Mode Catalog Query	43
Mode Selection Commands	43
Mode Selection Behavior	43
Radio Standard Profile	43
Radio Standard Retain Parameters	43
Radio Standard Signal Bandwidth	44
Restart Behavior	44
Trigger Mode Switching Behavior	44
Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)	45

Contents

Fetch, Measure & Read Commands for ACPower, MCPower	45
Average Count & Average /Hold Number	45
Average Mode	45
Carrier Integration Bandwidth, Width & Power Present	45
Initiate MCPower Measurement	46
Limit Test	46
Meas Method	47
Negative & Positive Offset Limits	47
Noise Correction State	47
Number of Carriers	48
Offset Filter Alpha & State	48
Offset Frequency	48
Offset Integration Bandwidth	49
Reference Carrier	49
Reference Carrier Frequency	49
Select Measurement	50
Total Power Setting	50
Burst Power Measurement (BPOwer)	51
Fetch, Measure & Read Commands	51
Average State	51
Average Type	51
Channel Power Measurement (CHPower)	52
Select Measurement	52
Average State	52
Integration Bandwidth	52
Complementary Cumulative Distribution Function Measurement (CCDF, PStatistic)	53
Fetch, Measure & Read Commands	53
Occupied Bandwidth Measurement (OBWidth)	54
Fetch, Measure & Read Commands	54
Spectrum Emission Mask Measurement (SEMask)	55
Fetch, Measure & Read Commands	55
Average State	55
Results Index	55
Sweep Time	55
Spurious Emissions Measurement (SPURious)	57
Detector Settings	57
Fast Spurious Measurement	57
5 Amplitude / Y Scale Commands	
Attenuation Query	59
Auto Scaling (Spurious Emissions Measurement)	60
Maximum Mixer Level	61

Power Reference Value (Spurious Emissions Measurement)	62
Preselector Adjust	63
Pre-Adjust for Min Clip	64
Reference Level	65
Reference Level Offset	66
Scale/Div	67
Scale Type	68
Y Axis Unit	69
6 Bandwidth Commands	
EMC Standard	71
Resolution Bandwidth	72
Video Bandwidth	73
7 Display Commands	
Display Enable	75
Full Screen Display	76
Window Annotation	77
Display Line Amplitude & State	78
8 Frequency Commands	
Center Freq	79
Start & Stop Frequencies	80
Frequency Offset	81
Frequency Span	82
Span/Resolution Bandwidth Ratio	83
Full Span & Last Span	84
Global Center Frequency	85
9 Input/Output Commands	
Analog Out	88
Aux IF Out	89
Correction Set Commands	90
External Preamp Gain	91
RF Calibrator Signal Selection	92
Input Mixer	92
10 Marker Commands	
General Differences	93
Position Markers vs. Value Markers	93
Band Power Functionality & Marker Modes	93
Band Power Settings & Marker Screen Positions	94
Marker Commands & PowerSuite Measurements	94
All Markers Off	95
Marker Type	95

Contents

Marker X Axis Value	96
Marker X Axis Position (in Trace Points)	97
Span Pair Commands	98
Delta Pair (Band Pair) Commands	99
Arbitrary Marker Pair functionality	99
Band Interval Left (Delta Pair Start Position)	100
Band/Interval Right (Delta Pair Stop Position)	101
Band/Interval Span	102
Band Span Auto	103
Continuous Peak Search Setting	104
Frequency Counter Gate Time	105
Frequency Counter State	106
Marker Delta to Span	107
Marker Maximum and Minimum (CCDF Measurement)	108
Marker Table	109
Marker to Reference Level	110
Marker Trace Auto	111
Marker Type (CCDF Measurement)	112
Marker Type (Spectrum Emission Mask Measurement)	113
Marker Type (Spurious Emissions Measurement)	114
Next Peak Commands	115
Peak Search, Min Search, Peak to Peak, Marker Trace	116
Signal Track	117
X Axis Scale (Readout)	118
Marker Function Settings	119
Direct Marker Positioning	119
Band Adjust	120
Detector Dwell Times	121
Measure at Marker Query	122
Preselector Automatic Centering	123
11 Span / X Scale Commands	
Scale/Div (CCDF Measurement)	125
Span (Channel Power Measurement)	125
Span (Occupied Bandwidth Measurement)	125
12 Sweep Commands	
External Gate Level	127
Gate Source	128
FFT Width	129
Sweep Points	130
Sweep Time	131
Sweep Time Rules	132

Sweep Type 133

13 Trace/Detector Commands

Detector	135
Command Parameter Mapping	135
Detector Auto Setting	137
Normalize Trace Data Settings (State, Reference Level, Reference Position)	137
Trace Mode (Type)	138
MAXHold & MINHold Issues	138
View & Blank Options	139
Trace Data	140
Measurement Setting	140
Data Format	140
Trace Copy & Exchange	141
Trace Math Add & Subtract	142
Mean Trace Data Query	143
Peak Data & Peak Points Queries	144
Trace Data Smooth & Points for Smoothing	145

14 Trigger Commands

Trigger Source	147
Trigger Delay	148
External 1, External 2 & Line Trigger Delay	149
External 1 Trigger Level	150
External 1, External 2 & Line Trigger Slope	151
RF Burst Trigger Level (Relative)	152
Video Trigger Delay	153
Video Trigger Level	154
Video Trigger Slope	155
Periodic Timer Sync Source	156

15 Preset Commands

Preset & Power-On Type	157
Mode Preset (MODE)	157
User Preset (USER)	157
Modes Affected by Preset	157
Data Affected by Preset	158
Last Preset (LAST)	158
Preset Type	159
IEEE 488.2 Common Command	160

16 System/Utility & IEEE 488.2 Common Commands

Align All Timeout	161
Auto Align	162

Contents

External Reference Output Control	163
License Display	164
RF Calibration Control	165
Show System	166
System Error Handling	167
System Language Query	168
System Power-On Type Options	169
Recall	170
Load from File	170
Amplitude Correction	170
Default Path & File Name	170
Save	171
Save to File	171
Amplitude Correction	171
Default Path & File Name	171
Frequency Reference Selection	172
Timebase DAC Calibration	173
IEEE 488.2 Common Commands	174
Clear Status	174
Operation Complete	174
Preset	174
Service Request Enable	174
17 Miscellaneous Commands	
External Mixer Presets	175
External Mixer Harmonic Value	176

Appendix A References

List of Commands

This list includes every SCPI command and query described in this document. To find a command in the list, search according to its first alphanumeric character, ignoring any leading ":" or "[" characters. The sole exception to this is the asterisk [*] prefix, identifying IEEE 488.2 Common commands and queries; all these appear at the start of the list.

This list includes commands and queries that are supported by PSA, by PXA, or by both instrument types. Clicking on any item in the list takes you to the section describing the command and defining which instruments support it. If a command does not appear in the list, this document does not identify any compatibility issues associated with it.

For conciseness and clarity, the entries below for each command or query do not include parameter lists (if any).

*CLS	174	
*OPC	174	
*OPC?	174	
*RST	160	
*SRE	174	
*SRE?	174	
:CALCulate:ACPower:LIMit:STATe	46	
:CALCulate:ACPower:LIMit:STATe?	46	
:CALCulate:ACPower:OFFSet:LIST:LIMit:NEGative[:UPPer]:DATA	47	
:CALCulate:ACPower:OFFSet:LIST:LIMit:NEGative[:UPPer]:DATA?	47	
:CALCulate:ACPower:OFFSet:LIST:LIMit:POSitive[:UPPer]:DATA	47	
:CALCulate:ACPower:OFFSet:LIST:LIMit:POSitive[:UPPer]:DATA?	47	
:CALCulate:LLINe[1]2 ... 6:AMPLitude:CMODE:RELative	36	
:CALCulate:LLINe[1]2 ... 6:AMPLitude:CMODE:RELative?	36	
:CALCulate:LLINe[1]2 ... 6:DATA	38	
:CALCulate:LLINe[1]2 ... 6:DATA?	38	
:CALCulate:LLINe[1]2 ... 6:DATA:MERGe	39	
:CALCulate:LLINe[1]2 ... 6:DISPlay	37	
:CALCulate:LLINe[1]2 ... 6:DISPlay?	37	
:CALCulate:LLINe[1]2 ... 6:FREQuency:CMODE:RELative	36	
:CALCulate:LLINe[1]2 ... 6:FREQuency:CMODE:RELative?	36	
:CALCulate:LLINe[1]2:CMODE	36	
:CALCulate:LLINe[1]2:CMODE?	36	
:CALCulate:LLINe[1]2:STATe	37	
:CALCulate:LLINe[1]2:STATe?	37	
:CALCulate:MAMarker:DETEctor[1]2 3	35	
:CALCulate:MAMarker:DETEctor[1]2 3?	35	
:CALCulate:MAMarker:DETEctor[1]2 3:DWELL	121	
:CALCulate:MAMarker:DETEctor[1]2 3:DWELL?	121	
:CALCulate:MAMarker:PCENter	123	
:CALCulate:MAMarker:PCENter?	123	
:CALCulate:MARKer[1]2 ... 12:CPSeArch[:STATe]	104	
:CALCulate:MARKer[1]2 ... 12:CPSeArch[:STATe]?	104	
:CALCulate:MARKer[1]2 ... 12:FCOunt[:STATe]	106	

List of Commands

:CALCulate:MARKer[1]2 ... 12:FCOunt[:STATe]?	106
:CALCulate:MARKer[1]2 ... 12:FCOunt:X?	106
:CALCulate:MARKer[1]2 ... 12:FUNction	119
:CALCulate:MARKer[1]2 ... 12:FUNction?	119
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:LEFT	100
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:LEFT?	100
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:RIGHT	101
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:RIGHT?	101
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:SPAN	102
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:SPAN?	102
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:SPAN:AUTO	103
:CALCulate:MARKer[1]2 ... 12:FUNction:BAND:SPAN:AUTO?	103
:CALCulate:MARKer[1]2 ... 12:FUNction:MAMarker?	122
:CALCulate:MARKer[1]2 ... 12:MAXimum	116
:CALCulate:MARKer[1]2 ... 12:MAXimum:LEFT	115
:CALCulate:MARKer[1]2 ... 12:MAXimum:NEXT	115
:CALCulate:MARKer[1]2 ... 12:MAXimum:RIGHT	115
:CALCulate:MARKer[1]2 ... 12:MINimum	116
:CALCulate:MARKer[1]2 ... 12:MODE	95
:CALCulate:MARKer[1]2 ... 12:MODE?	95
:CALCulate:MARKer[1]2 ... 12:PTPeak	116
:CALCulate:MARKer[1]2 ... 12[:SET]:DELta:SPAN	107
:CALCulate:MARKer[1]2 ... 12[:SET]:RLEVel	110
:CALCulate:MARKer[1]2 ... 12:STATe	95
:CALCulate:MARKer[1]2 ... 12:STATe?	95
:CALCulate:MARKer[1]2 ... 12:TRACe	116
:CALCulate:MARKer[1]2 ... 12:TRACe:AUTO	111
:CALCulate:MARKer[1]2 ... 12:TRACe:AUTO?	111
:CALCulate:MARKer[1]2 ... 12:X	96
:CALCulate:MARKer[1]2 ... 12:X?	96
:CALCulate:MARKer[1]2 ... 12:X:POSition	97
:CALCulate:MARKer[1]2 ... 12:X:POSition?	97
:CALCulate:MARKer[1]2 ... 12:X:READout	118
:CALCulate:MARKer[1]2 ... 12:X:READout?	118
:CALCulate:MARKer[1]2 ... 12:X:READout:AUTO	118
:CALCulate:MARKer[1]2 ... 12:X:READout:AUTO?	118
:CALCulate:MARKer[1]2 3 4:CPEak[:STATe]	104
:CALCulate:MARKer[1]2 3 4:CPEak[:STATe]?	104
:CALCulate:MARKer[1]2 3 4:MODE	95
:CALCulate:MARKer[1]2 3 4:MODE?	95
:CALCulate:MARKer[1]2 3 4:TRCKing[:STATe]	117
:CALCulate:MARKer[1]2 3 4:TRCKing[:STATe]?	117
:CALCulate:MARKer[1]2 3 4:X:CENTer	98
:CALCulate:MARKer[1]2 3 4:X:CENTer?	98

:CALCulate:MARKer[1]2 3 4:X:POSition:CENTer	119
:CALCulate:MARKer[1]2 3 4:X:POSition:CENTer?	119
:CALCulate:MARKer[1]2 3 4:X:POSition:SPAN	102
:CALCulate:MARKer[1]2 3 4:X:POSition:SPAN	119
:CALCulate:MARKer[1]2 3 4:X:POSition:SPAN?	102
:CALCulate:MARKer[1]2 3 4:X:POSition:SPAN?	119
:CALCulate:MARKer[1]2 3 4:X:POSition:START	100
:CALCulate:MARKer[1]2 3 4:X:POSition:START	119
:CALCulate:MARKer[1]2 3 4:X:POSition:START?	119
:CALCulate:MARKer[1]2 3 4:X:POSition:START?	100
:CALCulate:MARKer[1]2 3 4:X:POSition:STOP	101
:CALCulate:MARKer[1]2 3 4:X:POSition:STOP	119
:CALCulate:MARKer[1]2 3 4:X:POSition:STOP?	101
:CALCulate:MARKer[1]2 3 4:X:POSition:STOP?	119
:CALCulate:MARKer[1]2 3 4:X:SPAN	98
:CALCulate:MARKer[1]2 3 4:X:SPAN?	98
:CALCulate:MARKer[1]2 3 4:X:START	99
:CALCulate:MARKer[1]2 3 4:X:START?	99
:CALCulate:MARKer[1]2 3 4:X:STOP	99
:CALCulate:MARKer[1]2 3 4:X:STOP?	99
:CALCulate:MARKer[1-4]:FCOunt:GATetime	105
:CALCulate:MARKer[1-4]:FCOunt:GATetime?	105
:CALCulate:MARKer[1-4]:FCOunt:GATetime:AUTO	105
:CALCulate:MARKer[1-4]:FCOunt:GATetime:AUTO?	105
:CALCulate:MARKer:AOFF	95
:CALCulate:MARKer:TABLE[:STATe]	109
:CALCulate:MARKer:TABLE[:STATe]?	109
:CALCulate:MARKer:TRCKing[:STATe]	117
:CALCulate:MARKer:TRCKing[:STATe]?	117
:CALCulate:MCPower:OFFSet:LIST:LIMit:NEGative[:UPPer]:DATA	47
:CALCulate:MCPower:OFFSet:LIST:LIMit:NEGative[:UPPer]:DATA?	47
:CALCulate:MCPower:OFFSet:LIST:LIMit:POSitive[:UPPer]:DATA	47
:CALCulate:MCPower:OFFSet:LIST:LIMit:POSitive[:UPPer]:DATA?	47
:CALCulate:NTData[:STATe]	137
:CALCulate:NTData[:STATe]?	137
:CALCulate:PSTatistic:MARKer[1]2 ... 12:MODE	112
:CALCulate:PSTatistic:MARKer[1]2 3 4:MAXimum	108
:CALCulate:PSTatistic:MARKer[1]2 3 4:MINimum	108
:CALCulate:PSTatistic:MARKer[1]2 3 4:MODE	112
:CALCulate:PSTatistic:MARKer[1]2 3 4[:STATe]	112
:CALCulate:PSTatistic:MARKer[1]2 3 4[:STATe]?	112
:CALCulate:SEMAsk:MARKer[1]2 ... 12:MODE	113
:CALCulate:SEMAsk:MARKer[1]2 ... 12:MODE?	113
:CALCulate:SEMAsk:MARKer[1]2 3 4[:STATe]	113

List of Commands

:CALCulate:SEMAsk:MARKer[1]2 3 4[:STATe]?	113
:CALCulate:SPURious:MARKer[1]2 ... 12:MODE	114
:CALCulate:SPURious:MARKer[1]2 ... 12:MODE?	114
:CALCulate:SPURious:MARKer[1]2 3 4[:STATe]	114
:CALCulate:SPURious:MARKer[1]2 3 4[:STATe]?	114
:CALibration[:ALL]?	161
:CALibration:AUTO	162
:CALibration:AUTO?	162
:CALibration:FREQuency:REFeRence:COARse	173
:CALibration:FREQuency:REFeRence:COARse?	173
:CALibration:FREQuency:REFeRence:FINE	173
:CALibration:FREQuency:REFeRence:FINE?	173
:CALibration:SOURce:STATe	165
:CALibration:SOURce:STATe?	165
:CONFigure?	42
:CONFigure:ACPower	50
:CONFigure:ACPower BPOWER CHPower EBWidth OBWidth PSTatistic SANalyzer SEMAsk SPURious	42
:CONFigure:CHPower	52
:CONFigure:MCPower	50
:CONFigure:SANalyzer	29
:DISPlay:<meas>:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:PDIVision	67
:DISPlay:<meas>:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:RLEVel	65
:DISPlay:ENABLE	75
:DISPlay:ENABLE?	75
:DISPlay:FSCReen[:STATe]	76
:DISPlay:FSCReen[:STATe]?	76
:DISPlay:MENU[:STATe]	76
:DISPlay:PSTatistic:VIEW[1]:WINDow2:TRACe:X[:SCALe]:PDIVision	125
:DISPlay:PSTatistic:VIEW[1]:WINDow2:TRACe:X[:SCALe]:PDIVision?	125
:DISPlay:PSTatistic:XSCale	125
:DISPlay:PSTatistic:XSCale?	125
:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:COUPle	60
:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:COUPle?	60
:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:RLEVel	62
:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALe]:RLEVel?	62
:DISPlay:WINDow[1]:ANNotation[:ALL]	77
:DISPlay:WINDow[1]:ANNotation[:ALL]?	77
:DISPlay:WINDow[1]:TRACe:X[:SCALe]:OFFSet	81
:DISPlay:WINDow[1]:TRACe:Y:DLINe	78
:DISPlay:WINDow[1]:TRACe:Y:DLINe:STATe	78
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:NRLevel	137
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:NRLevel?	137
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:NRPosition	137
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:NRPosition?	137

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:PDIVision	67
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel	65
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel?	65
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet	66
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?	66
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:SPACing	68
:FETCh:ACPower?	45
:FETCh:MCPower[n]?	45
:FETCh:OBWidth?	54
:FETCh:OBWidth:FERRor?	54
:FETCh:OBWidth[n]?	54
:FETCh:OBWidth:OBWidth?	54
:FETCh:OBWidth:XDB?	54
:FETCh:PStatistic[n]?	53
:FETCh:SEMAsk[n]?	55
:FETCh:TXPower BPOWer[n]?	51
:GLOBal:FREQuency:CENTer[:STATe]	85
:GLOBal:FREQuency:CENTer[:STATe]?	85
:INITiate:ACPower	46
:INITiate:CONTInuous	44
:INITiate:CONTInuous?	44
:INITiate[:IMMediate]	44
:INITiate:MCPower	46
:INITiate:REStart	44
:INPut:MIXer	92
:INPut:MIXer?	92
:INPut:MIXer:TYPE	92
:INPut:MIXer:TYPE?	92
:INSTrument:CATalog?	43
:INSTrument:COUPle:FREQuency:CENTer	85
:INSTrument:COUPle:FREQuency:CENTer?	85
:INSTrument[:SELect]	43
:INSTrument[:SELect]?	43
:MEASure:ACPower?	45
:MEASure:EMl:MARKer[1]2[...] 12?	122
:MEASure:MCPower[n]?	45
:MEASure:OBWidth?	54
:MEASure:OBWidth:FERRor?	54
:MEASure:OBWidth[n]?	54
:MEASure:OBWidth:OBWidth?	54
:MEASure:OBWidth:XDB?	54
:MEASure:PStatistic[n]?	53
:MEASure:SEMAsk[n]?	55
:MEASure:TXPower BPOWer[n]?	51

List of Commands

:MMEMory:LOAD:CORRection	170
:MMEMory:LOAD:STATe	170
:MMEMory:STORe:CORRection	171
:MMEMory:STORe:STATe	171
:OUTPut:ANALog	88
:OUTPut:ANALog?	88
:OUTPut:ANALog:AUTO	88
:OUTPut:ANALog:AUTO?	88
:OUTPut:AUX	89
:OUTPut:AUX?	89
:READ:ACPower?	45
:READ:MCPower[n]?	45
:READ:OBWidth?	54
:READ:OBWidth:FERRor?	54
:READ:OBWidth[n]?	54
:READ:OBWidth:OBWidth?	54
:READ:OBWidth:XDB?	54
:READ:PStatistic[n]?	53
:READ:SEMask[n]?	55
:READ:TXPower BPOWer[n]?	51
[:SENSe]:<meas>:FREQuency:SPAN	80
[:SENSe]:<meas>:FREQuency:SYNThesis:AUTO[:STATe]	40
[:SENSe]:<meas>:FREQuency:SYNThesis:AUTO[:STATe]?	40
[:SENSe]:<meas>:FREQuency:SYNThesis[:STATe]	40
[:SENSe]:<meas>:FREQuency:SYNThesis[:STATe]?	40
[:SENSe]:<meas>:TRIGger:SOURce	147
[:SENSe]:ACPower:AVERage[:STATe]	45
[:SENSe]:ACPower:AVERage[:STATe]?	45
[:SENSe]:ACPower:CARRier[1]2:COUNT	48
[:SENSe]:ACPower:CARRier[1]2:COUNT?	48
[:SENSe]:ACPower:CARRier[1]2:LIST:BANDwidth BWIDTH[:INTegration]	46
[:SENSe]:ACPower:CARRier[1]2:LIST:BANDwidth BWIDTH[:INTegration]?	46
[:SENSe]:ACPower:CARRier[1]2:LIST:PPResent	46
[:SENSe]:ACPower:CARRier[1]2:LIST:PPResent?	46
[:SENSe]:ACPower:CARRier[1]2:LIST:WIDTH	46
[:SENSe]:ACPower:CARRier[1]2:LIST:WIDTH?	46
[:SENSe]:ACPower:CARRier[1]2:RCARRier	49
[:SENSe]:ACPower:CARRier[1]2:RCARRier?	49
[:SENSe]:ACPower:CARRier[1]2:RCARRier:AUTO	49
[:SENSe]:ACPower:CARRier[1]2:RCARRier:AUTO?	49
[:SENSe]:ACPower:CARRier[1]2:RCFRequency	49
[:SENSe]:ACPower:CARRier[1]2:RCFRequency?	49
[:SENSe]:ACPower:CARRier[1]2:RCFRequency:AUTO	49
[:SENSe]:ACPower:CARRier[1]2:RCFRequency:AUTO?	49

[:SENSe]:ACPower:CORRection:NOISe[:AUTO]	48
[:SENSe]:ACPower:FILTer[:RRC]:ALPHa	48
[:SENSe]:ACPower:FILTer[:RRC][:STATe]	48
[:SENSe]:ACPower:LIMit[:STATe]	46
[:SENSe]:ACPower:LIMit[:STATe]?	46
[:SENSe]:ACPower:METHod	47
[:SENSe]:ACPower:METHod?	47
[:SENSe]:ACPower:OFFSet[1]2:LIST:BANDwidth BWIDth[:INTegration]	49
[:SENSe]:ACPower:OFFSet[1]2:LIST:BANDwidth BWIDth[:INTegration]?	49
[:SENSe]:ACPower:OFFSet[1]2:LIST[:FREQuency]	48
[:SENSe]:ACPower:OFFSet[1]2:LIST[:FREQuency]?	48
[:SENSe]:ACPower:OFFSet[1]2:LIST:RCARrier	47
[:SENSe]:ADC:DITHer:AUTO[:STATe]	30
[:SENSe]:ADC:DITHer:AUTO[:STATe]?	30
[:SENSe]:ADC:DITHer[:STATe]	30
[:SENSe]:ADC:DITHer[:STATe]?	30
[:SENSe]:ADC:RANGe	31
[:SENSe]:ADC:RANGe?	31
[:SENSe]:AVERAge:COUNT	32
[:SENSe]:AVERAge:COUNT?	32
[:SENSe]:AVERAge[:STATe]	138
[:SENSe]:AVERAge[:STATe]	33
[:SENSe]:AVERAge[:STATe]?	138
[:SENSe]:AVERAge[:STATe]?	33
[:SENSe]:AVERAge:TYPE	34
[:SENSe]:AVERAge:TYPE?	34
[:SENSe]:AVERAge:TYPE:AUTO	34
[:SENSe]:AVERAge:TYPE:AUTO?	34
[:SENSe]:BANDwidth BWIDth[:RESolution]	72
[:SENSe]:BANDwidth BWIDth[:RESolution]?	72
[:SENSe]:BANDwidth BWIDth[:RESolution]:AUTO	72
[:SENSe]:BANDwidth BWIDth[:RESolution]:AUTO?	72
[:SENSe]:BANDwidth BWIDth[:RESolution]:MODE	71
[:SENSe]:BANDwidth BWIDth[:RESolution]:MODE?	71
[:SENSe]:BANDwidth BWIDth:VIDeo	73
[:SENSe]:BANDwidth BWIDth:VIDeo?	73
[:SENSe]:BANDwidth BWIDth:VIDeo:AUTO	73
[:SENSe]:BANDwidth BWIDth:VIDeo:AUTO?	73
[:SENSe]:BPOWer TXPower:AVERAge[:STATe]	51
[:SENSe]:BPOWer TXPower:AVERAge[:STATe]?	51
[:SENSe]:BPOWer:AVERAge:TYPE	51
[:SENSe]:BPOWer:AVERAge:TYPE?	51
[:SENSe]:CHPower:AVERAge[:STATe]	52
[:SENSe]:CHPower:AVERAge[:STATe]?	52

List of Commands

[:SENSe]:CHPower:BANDwidth BWIDth:INTegration	52
[:SENSe]:CHPower:FREQuency:SPAN	125
[:SENSe]:CORRection:CSET[1]2 ... 6:DATA	90
[:SENSe]:CORRection:CSET[1]2 ... 6:DATA?	90
[:SENSe]:CORRection:CSET[1]2 ... 6:DATA:MERGe	90
[:SENSe]:CORRection:CSET[1]2 ... 6:DELeTe	90
[:SENSe]:CORRection:CSET[1]2 ... 6[:STATe]	90
[:SENSe]:CORRection:CSET[1]2 ... 6[:STATe]?	90
[:SENSe]:CORRection:CSET[1]2 ... 6:X:SPACing	90
[:SENSe]:CORRection:CSET[1]2 ... 6:X:SPACing?	90
[:SENSe]:CORRection:CSET:ALL:DELeTe	90
[:SENSe]:CORRection:CSET:ALL[:STATe]	90
[:SENSe]:CORRection:CSET:ALL[:STATe]?	90
[:SENSe]:CORRection:OFFSet[:MAGNitude]	91
[:SENSe]:CORRection:OFFSet[:MAGNitude]?	91
[:SENSe]:CORRection:SA[:RF]:GAIN	91
[:SENSe]:CORRection:SA[:RF]:GAIN?	91
[:SENSe]:DETEctor:AUTO	137
[:SENSe]:DETEctor:AUTO?	137
[:SENSe]:DETEctor[:FUNction]	135
[:SENSe]:DETEctor[:FUNction]?	135
[:SENSe]:DETEctor:TRACe[1]2 ...6	135
[:SENSe]:DETEctor:TRACe[1]2 ...6?	135
[:SENSe]:DETEctor:TRACe[1]2 ...6:AUTO	137
[:SENSe]:DETEctor:TRACe[1]2 ...6:AUTO?	137
[:SENSe]:EMI:MEASure:DETEctor:AVERage[:STATe]	35
[:SENSe]:EMI:MEASure:DETEctor:DWELL	121
[:SENSe]:EMI:MEASure:DETEctor:QPEak[:STATe]	35
[:SENSe]:EMI:MEASure:PCENter[:STATe]	123
[:SENSe]:EMI:MEASure:PCENter[:STATe]?	123
[:SENSe]:FEED	92
[:SENSe]:FEED?	92
[:SENSe]:FREQuency:CENTer	79
[:SENSe]:FREQuency:OFFSet	81
[:SENSe]:FREQuency:OFFSet?	81
[:SENSe]:FREQuency:SPAN	82
[:SENSe]:FREQuency:SPAN?	82
[:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio	83
[:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?	83
[:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO	83
[:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO?	83
[:SENSe]:FREQuency:SPAN:FULL	84
[:SENSe]:FREQuency:SPAN:PREVious	84
[:SENSe]:FREQuency:START	80

[:SENSe]:FREQuency:START	82	
[:SENSe]:FREQuency:START?	80	
[:SENSe]:FREQuency:STOP	80	
[:SENSe]:FREQuency:STOP	82	
[:SENSe]:FREQuency:STOP?	80	
[:SENSe]:FREQuency:SYNThesis[:STATe]	40	
[:SENSe]:FREQuency:SYNThesis[:STATe]?	40	
[:SENSe]:FREQuency:SYNThesis:AUTO[:STATe]	40	
[:SENSe]:FREQuency:SYNThesis:AUTO[:STATe]?	40	
[:SENSe]:IF:GAIN:FFT:AUTO[:STATe]	31	
[:SENSe]:IF:GAIN:FFT:AUTO[:STATe]?	31	
[:SENSe]:MCPower:AVERAge[:STATe]	45	
[:SENSe]:MCPower:AVERAge[:STATe]?	45	
[:SENSe]:MCPower:AVERAge:TCONtrol	45	
[:SENSe]:MCPower:AVERAge:TCONtrol?	45	
[:SENSe]:MCPower:CARRier:AUTO[:STATe]	50	
[:SENSe]:MCPower:CARRier:AUTO[:STATe]?	50	
[:SENSe]:MCPower:CARRier:COUNt	48	
[:SENSe]:MCPower:CARRier:COUNt?	48	
[:SENSe]:MCPower:CARRier:LIST:BANDwidth BWiDth[:INTegration]	46	
[:SENSe]:MCPower:CARRier:LIST:BANDwidth BWiDth[:INTegration]?	46	
[:SENSe]:MCPower:CARRier:LIST:PPResent	46	
[:SENSe]:MCPower:CARRier:LIST:PPResent?	46	
[:SENSe]:MCPower:CARRier:LIST:WiDth	46	
[:SENSe]:MCPower:CARRier:LIST:WiDth?	46	
[:SENSe]:MCPower:CARRier[:POWer]	50	
[:SENSe]:MCPower:CARRier[:POWer]?	50	
[:SENSe]:MCPower:CORRection:NOISe[:AUTO]	47	
[:SENSe]:MCPower:CORRection:NOISe[:AUTO]?	47	
[:SENSe]:MCPower:FILTer[:RRC]:ALPHa	48	
[:SENSe]:MCPower:FILTer[:RRC][:STATe]	48	
[:SENSe]:MCPower:LIMit[:STATe]	46	
[:SENSe]:MCPower:LIMit[:STATe]?	46	
[:SENSe]:MCPower:METhod	47	
[:SENSe]:MCPower:METhod?	47	
[:SENSe]:MCPower:OFFSet:LIST:BANDwidth[:INTegration]	49	
[:SENSe]:MCPower:OFFSet:LIST:BANDwidth[:INTegration]?	49	
[:SENSe]:MCPower:OFFSet:LIST[:FREQuency]	48	
[:SENSe]:MCPower:OFFSet:LIST[:FREQuency]?	48	
[:SENSe]:MCPower:RCARrier	49	
[:SENSe]:MCPower:RCARrier?	49	
[:SENSe]:MCPower:RCARrier:AUTO	49	
[:SENSe]:MCPower:RCARrier:AUTO?	49	
[:SENSe]:MCPower:RCFRequency	49	

List of Commands

[.SENSe]:MCPower:RCFRequency?	50
[.SENSe]:MCPower:RCFRequency:AUTO	50
[.SENSe]:MCPower:RCFRequency:AUTO?	50
[.SENSe]:MIXer:BAND	175
[.SENSe]:MIXer:BAND?	175
[.SENSe]:MIXer:HARMonic	176
[.SENSe]:MIXer:HARMonic?	176
[.SENSe]:MIXer:HARMonic:AUTO	176
[.SENSe]:MIXer:HARMonic:AUTO?	176
[.SENSe]:OBWidth:FREQuency:SPAN	125
[.SENSe]:OBWidth:FREQuency:SPAN?	125
[.SENSe]:POWer[:RF]:ATTenuation?	59
[.SENSe]:POWer[:RF]:MIXer:RANGe[:UPPer]	61
[.SENSe]:POWer[:RF]:MIXer:RANGe[:UPPer]?	61
[.SENSe]:POWer[:RF]:MMW:PADJust	63
[.SENSe]:POWer[:RF]:MW:PADJust	63
[.SENSe]:POWer[:RF]:PADJust	63
[.SENSe]:POWer[:RF]:PADJust?	63
[.SENSe]:POWer[:RF]:PADJust:PRESelector	63
[.SENSe]:POWer[:RF]:PADJust:PRESelector?	63
[.SENSe]:POWer[:RF]:RANGe:AUTO	64
[.SENSe]:POWer[:RF]:RANGe:AUTO?	64
[.SENSe]:POWer[:RF]:RANGe:OPTimize:ATTenuation	64
[.SENSe]:POWer[:RF]:RANGe:OPTimize:ATTenuation?	64
[.SENSe]:PStatistic:MARKer[1][2][3][4]:MAXimum	108
[.SENSe]:PStatistic:MARKer[1][2][3][4]:MINimum	108
[.SENSe]:RADio:STANdard:PROFile	43
[.SENSe]:RADio:STANdard:RPARams[:STATe]	43
[.SENSe]:RADio:STANdard:RPARams[:STATe]?	43
[.SENSe]:RADio:STANdard:SBWidth	44
[.SENSe]:RADio:STANdard:SBWidth?	44
[.SENSe]:ROSCillator:OUTPut[:STATe]	163
[.SENSe]:ROSCillator:OUTPut[:STATe]?	163
[.SENSe]:ROSCillator:SOURce	172
[.SENSe]:ROSCillator:SOURce?	172
[.SENSe]:ROSCillator:SOURce:TYPE	172
[.SENSe]:ROSCillator:SOURce:TYPE?	172
[.SENSe]:SEMask:AVERAge[:STATe]	55
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEp[:TIME]	56
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEptime	56
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEp[:TIME]?	56
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEptime?	56
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEp[:TIME]:AUTO	56
[.SENSe]:SEMask:OFFSet[1][2]:LIST:SWEEptime:AUTO	56

[:SENSe]:SEMask:OFFSet[1]2:LIST:SWEEp[:TIME]:AUTO?	56
[:SENSe]:SEMask:OFFSet[1]2:LIST:SWEEptime:AUTO?	56
[:SENSe]:SEMask:RESult	55
[:SENSe]:SPURious:FSMeas	57
[:SENSe]:SPURious:FSMeas?	57
[:SENSe]:SPURious:POWer[:RF]:LEVel	62
[:SENSe]:SPURious:POWer[:RF]:LEVel?	62
[:SENSe]:SPURious:POWer[:RF]:RANGe:AUTO	60
[:SENSe]:SPURious:POWer[:RF]:RANGe:AUTO?	60
[:SENSe]:SPURious[:RANGe][:LIST]:DETector[1]2[:FUNcTion]?	57
[:SENSe]:SPURious[:RANGe][:LIST]:DETector[1][:FUNcTion]	57
[:SENSe]:SPURious[:RANGe][:LIST]:DETector[2]?	57
[:SENSe]:SPURious[:RANGe][:LIST]:DETector2[:FUNcTion]	57
[:SENSe]:SPURious[:RANGe][:LIST]:DETector[:FUNcTion]	57
[:SENSe]:SPURious:REPT:MODE	57
[:SENSe]:SPURious:REPT:MODE?	57
[:SENSe]:SWEEp:EGATe:EXTeRnal[1]2:LEVel	127
[:SENSe]:SWEEp:EGATe:EXTeRnal[1]2:LEVel?	127
[:SENSe]:SWEEp:EGATe:SOURce	128
[:SENSe]:SWEEp:EGATe:SOURce?	128
[:SENSe]:SWEEp:FFT:SPAN:RATio	129
[:SENSe]:SWEEp:FFT:SPAN:RATio?	129
[:SENSe]:SWEEp:FFT:WIDTh	129
[:SENSe]:SWEEp:FFT:WIDTh?	129
[:SENSe]:SWEEp:POINts	130
[:SENSe]:SWEEp:POINts?	130
[:SENSe]:SWEEp:TIME	131
[:SENSe]:SWEEp:TIME?	131
[:SENSe]:SWEEp:TIME:AUTO	131
[:SENSe]:SWEEp:TIME:AUTO?	131
[:SENSe]:SWEEp:TIME:AUTO:MODE	132
[:SENSe]:SWEEp:TIME:AUTO:MODE?	132
[:SENSe]:SWEEp:TIME:AUTO:RULes	132
[:SENSe]:SWEEp:TIME:AUTO:RULes?	132
[:SENSe]:SWEEp:TYPE	133
[:SENSe]:SWEEp:TYPE	133
[:SENSe]:SWEEp:TYPE?	133
[:SENSe]:SWEEp:TYPE:AUTO	133
[:SENSe]:SWEEp:TYPE:AUTO?	133
[:SENSe]:TXPower:AVERAge:TYPE	51
[:SENSe]:TXPower:AVERAge:TYPE?	51
:SYSTem:CONFIgure:LKEY:STATe	164
:SYSTem:CONFIgure:LKEY:STATe?	164
:SYSTem:ERRor[:NEXT]?	167

List of Commands

:SYSTem:LANGUage?	168
:SYSTem:PON:TYPE	157
:SYSTem:PON:TYPE?	157
:SYSTem:PRESet:TYPE	159
:SYSTem:PRESet:TYPE?	159
:TRACe[1]2 ... 6:DISPlay[:STATe]	139
:TRACe[1]2 ... 6:DISPlay[:STATe]?	139
:TRACe[1]2 ... 6:MODE	138
:TRACe[1]2 ... 6:MODE?	138
:TRACe[1]2 ... 6:UPDate[:STATe]	139
:TRACe[1]2 ... 6:UPDate[:STATe]?	139
:TRACe[1]2 ...6:TYPE	138
:TRACe[1]2 ...6:TYPE?	138
:TRACe[1]2 3:COPIY	141
:TRACe[1]2 3:COPIY?	141
:TRACe[1]2 3[:DATA]	140
:TRACe[1]2 3[:DATA]?	140
:TRACe[1]2 3[:DATA]?	140
:TRACe[1]2 3:EXCHange	141
:TRACe[1]2 3:EXCHange?	141
:TRACe[1]2 3:MATH:ADD	142
:TRACe[1]2 3:MATH:MEAN?	143
:TRACe[1]2 3:MATH:PEAK[:DATA]?	144
:TRACe[1]2 3:MATH:PEAK:POINts?	144
:TRACe[1]2 3:MATH:SMOoth	145
:TRACe[1]2 3:MATH:SMOoth:POINts	145
:TRACe[1]2 3:MATH:SMOoth:POINts?	145
:TRACe[1]2 3:MATH:SUBTract	142
:TRACe[1]2 3:MATH:SUBTract:DLINe	142
:TRIGger:<meas>[:SEQuence]:SOURce	147
:TRIGger:<meas>[:SEQuence]:SOURce?	147
:TRIGger[:SEQuence]:DELay	148
:TRIGger[:SEQuence]:DELay?	148
:TRIGger[:SEQuence]:DELay:STATe	148
:TRIGger[:SEQuence]:DELay:STATe?	148
:TRIGger[:SEQuence]:EXTernal1 2:DELay	149
:TRIGger[:SEQuence]:EXTernal1 2:DELay?	149
:TRIGger[:SEQuence]:EXTernal1 2:DELay:STATe	149
:TRIGger[:SEQuence]:EXTernal1 2:DELay:STATe?	149
:TRIGger[:SEQuence]:EXTernal1 2:SLOPe	151
:TRIGger[:SEQuence]:EXTernal1 2:SLOPe?	151
:TRIGger[:SEQuence]:EXTernal1:LEVel	150
:TRIGger[:SEQuence]:EXTernal1:LEVel?	150
:TRIGger[:SEQuence]:EXTernal:DELay	149

:TRIGger[:SEQuence]:FRAMe:SYNc	156
:TRIGger[:SEQuence]:FRAMe:SYNc?	156
:TRIGger[:SEQuence]:IF:DELay	148
:TRIGger[:SEQuence]:IF:LEVel	154
:TRIGger[:SEQuence]:IF:LEVel?	154
:TRIGger[:SEQuence]:IF:SLOPe	155
:TRIGger[:SEQuence]:IF:SLOPe?	155
:TRIGger[:SEQuence]:LINE:DELay	149
:TRIGger[:SEQuence]:LINE:DELay?	149
:TRIGger[:SEQuence]:LINE:DELay:STATe	149
:TRIGger[:SEQuence]:LINE:DELay:STATe?	149
:TRIGger[:SEQuence]:LINE:SLOPe	151
:TRIGger[:SEQuence]:LINE:SLOPe?	151
:TRIGger[:SEQuence]:OFFSet	153
:TRIGger[:SEQuence]:OFFSet?	153
:TRIGger[:SEQuence]:OFFSet:STATe	153
:TRIGger[:SEQuence]:OFFSet:STATe?	153
:TRIGger[:SEQuence]:RFBurst:LEVel	152
:TRIGger[:SEQuence]:RFBurst:LEVel:RELative	152
:TRIGger[:SEQuence]:RFBurst:LEVel:RELative?	152
:TRIGger[:SEQuence]:SLOPe	155
:TRIGger[:SEQuence]:SLOPe?	155
:TRIGger[:SEQuence]:SOURce	147
:TRIGger[:SEQuence]:SOURce?	147
:TRIGger[:SEQuence]:VIDeo:DELay	153
:TRIGger[:SEQuence]:VIDeo:DELay?	153
:TRIGger[:SEQuence]:VIDeo:DELay:STATe	153
:TRIGger[:SEQuence]:VIDeo:DELay:STATe?	153
:TRIGger[:SEQuence]:VIDeo:LEVel	154
:TRIGger[:SEQuence]:VIDeo:LEVel?	154
:TRIGger[:SEQuence]:VIDeo:SLOPe	155
:TRIGger[:SEQuence]:VIDeo:SLOPe?	155
:UNIT:POWer	69

List of Commands

1 Using This Guide

The purpose of this document is to help you convert SCPI remote control code written for the Agilent PSA instrument to code that is compatible with the Keysight PXA instrument. The document describes differences in the syntax and behavior of commands, explains the aliasing of certain PSA commands to PXA equivalents, and specifies which PSA commands are not supported by PXA.

This document's emphasis is on SCPI code compatibility issues, so, in general, differences of front panel keys and other features of the instrument hardware are not discussed here, except where such information is required to explain programming issues.

To locate a SCPI command or query, refer to the chapter [“List of Commands” on page 11](#), which includes commands that may be used in PSA, PXA or in both instruments. For details of how to find measurement-specific commands, see [“Measurement-Specific Commands” on page 26](#).

TIP If a command or query is not described anywhere in this document, you can assume that it is supported by both PSA and PXA without any known compatibility issues.

Command Compatibility Status Icons

Each of the PSA SCPI commands and queries described in this manual has been assigned a status to indicate its compatibility with PXA. The status of each command is indicated by an icon as follows:

Icon	Status	Meaning
	Compatible	This is either a native PXA command, or it is a PSA command that is 100% compatible with PXA.
	Modified	This PSA command may be sent to PXA, but there are some differences in behavior or response that may affect the operation of programs using this command that were written for PSA.
	Not supported	This PSA command is not supported by PXA. In programs written for PSA, you must replace the command with an equivalent PXA-compatible command or sequence.

Organization of this Manual

The document is divided as follows:

Using This Guide

Measurement-Specific Commands

- [Using This Guide](#) (this chapter).
- [“General Issues” on page 27](#). Describes issues that relate to all modes and measurements.
- [“Swept SA Measurement Setup” on page 29](#). Describes commands that relate to the Swept SA measurement.
- [“PowerSuite Measurements Setup” on page 41](#). Describes commands that relate to specific measurements in PowerSuite.
- [“Amplitude / Y Scale Commands” on page 59](#). Describes commands that relate to Amplitude or Y Scale functionality.
- [“Bandwidth Commands” on page 71](#). Describes commands that relate to Bandwidth functionality.
- [“Display Commands” on page 75](#). Describes commands that relate to Display functionality.
- [“Frequency Commands” on page 79](#). Describes commands that relate to Frequency functionality.
- [“Input/Output Commands” on page 87](#). Describes commands that relate to the instrument’s Input/Output functionality.
- [“Marker Commands” on page 93](#). Describes commands that relate to the instrument marker system.
- [“Span / X Scale Commands” on page 125](#). Describes commands that relate to the Span or X Scale functionality.
- [“Sweep Commands” on page 127](#). Describes commands that relate to the Sweep functionality.
- [“Trace/Detector Commands” on page 135](#). Describes commands that relate to the instrument trace and detector functionality.
- [“Trigger Commands” on page 147](#). Describes commands that relate to trigger operation.
- [“Preset Commands” on page 157](#). Describes commands that relate to instrument preset.
- [“System/Utility & IEEE 488.2 Common Commands” on page 161](#). Describes commands that relate to items in the instrument System/Utility menu. This chapter also includes issues relating to IEEE 488.2 Common commands (that is, those commands prefixed with an asterisk [*]).
- [“Miscellaneous Commands” on page 175](#). Describes commands that do not fit into one of the groups above.

Measurement-Specific Commands

In general, this guide is organized by command function, rather than by measurement. For example, if you want to look up an Amplitude-related function that’s specific to the Spurious Emissions measurement, you should look in the chapter [“Amplitude / Y Scale Commands” on page 59](#). The exceptions to this organization are Measurement Setup commands, which are split between [“Swept SA Measurement Setup” on page 29](#) and [“PowerSuite Measurements Setup” on page 41](#).

2 General Issues

Numeric Data Formatting

For numeric data with values greater than or equal to zero, PSA prefixes a "+" sign.

PXA does not affix a "+" sign to data in these cases.

Sensitivity to Input Power Level

The ideal setting of the input attenuation depends on the trade-off between the noise floor and large signal performance such as distortion or compression. The warranted specifications of PXA and PSA are very similar for these items, but at some frequencies the experimentally determined optimum attenuation may differ. It is not uncommon for PXA to require 2 dB or 4 dB more attenuation, or 2 dB less, than PSA for optimum compatibility.

Band Break Location

PSA switches to preselected behavior at 3 GHz, while PXA switches at 3.6 GHz (except when pre-selector bypass is enabled). Below the switch point, both analyzers perform at their best for amplitude accuracy and wideband IF performance; above the switch point, they perform at their best for harmonic distortion. PXA is not compatible with PSA in cases where the dynamic range required for harmonic distortion measurements between 3 GHz and 3.6 GHz exceeds PXA performance, but not PSA performance.

General Issues
Band Break Location

3 Swept SA Measurement Setup

Switching to Spectrum Analyzer Measurement



`:CONFigure:SANalyzer`

- On receipt of this command, PSA switches the measurement only, while PXA also **presets** settings to their default values. Thus, for PXA, you must re-initialize settings such as number of points or RBW/VBW ratio.

Alternatively, you can use `CONFigure:SANalyzer:NDEFault` to switch the measurement without presetting to defaults.

- PSA turns off markers, but PXA does not. For example, if you send `CALCulate:MARKer[...]:STATE ON` after `CONFigure:SANalyzer`, PSA always puts the marker at the center of the screen, while PXA may keep an existing marker on screen without moving it to the center.
- With PSA, if trigger mode is in **Single**, sending `:CONFigure:SANalyzer` initiates the trigger. PXA does not.

Swept SA Measurement Setup

ADC Dither

ADC Dither

PXA commands and query:

 [:SENSe]:ADC:DITHer[:STATe] OFF|ON|HIGH

 [:SENSe]:ADC:DITHer:AUTO[:STATe] OFF|ON|0|1

 [:SENSe]:ADC:DITHer:AUTO[:STATe]?

PSA command:

 [:SENSe]:ADC:DITHer[:STATe] OFF|ON|AUTO

PXA and PSA query:

 [:SENSe]:ADC:DITHer[:STATe]?

The PSA command [:SENSe]:ADC:DITHer AUTO is aliased in PXA to [:SENSe]:ADC:DITHer:AUTO[:STATe] ON. Therefore, even in PXA, the state variable cannot be a true Boolean, so the query [:SENSe]:ADC:DITHer[:STATe]? returns OFF or ON (not 1 or 0 like a true Boolean).

In PSA, the controls for this functionality are under the **Auto Couple** key, and the functionality is available in all measurements.

In PXA, only the Swept SA measurement supports this functionality.

ADC Range & FFT IF Gain Auto

PXA command and query:

 [:SENSe]:IF:GAIN:FFT:AUTO[:STATe] OFF|ON|0|1

 [:SENSe]:IF:GAIN:FFT:AUTO[:STATe]?

PSA command and query supported in PXA for backwards compatibility:

 [:SENSe]:ADC:RANGe AUTO|NONE

 [:SENSe]:ADC:RANGe?

In PXA, this command and query are accepted without error, but ignored.

Swept SA Measurement Setup Average Count & Average /Hold Number

Average Count & Average /Hold Number

PSA and PXA command and query:

 [:SENSe]:AVERAge:COUNT <integer>

 [:SENSe]:AVERAge:COUNT?

In PSA, this parameter is measurement local, and is coupled with [:SENSe]:<meas>:AVERAge:COUNT.

In PSA's CCDF and Spurious Emissions measurements, this value is reset to its default on measurement switching.

In PSA, when changing the **Average Count** (now **Average/Hold Number**), you must restart the trace at the beginning of a sweep to ensure valid average data. In PXA, the system automatically ensures valid results when changing the count limit.

Average State

PSA and PXA command and query:

 [:SENSe]:AVERAge[:STATe] ON|OFF|1|0

 [:SENSe]:AVERAge[:STATe]?

In PSA, this parameter is affected by the Retain Parameter setting. When Retain Parameter is on, the value is retained as local value at measurement switch. When Retain Parameter is off, the value is reset to its default value on a measurement switch.

In PXA, this functionality is supported as a measurement local parameter.

Swept SA Measurement Setup

Average Type

Average Type

PSA and PXA commands and queries:

 [:SENSe] :AVERAge :TYPE RMS | LOG | SCALAr

 [:SENSe] :AVERAge :TYPE?

 [:SENSe] :AVERAge :TYPE :AUTO OFF | ON | 0 | 1

 [:SENSe] :AVERAge :TYPE :AUTO?

In PSA, this parameter is affected by the Retain Parameter setting for most measurements. When Retain Parameter is on, the value is retained as local value at measurement switch. When Retain Parameter is off, the value is reset to its default value on a measurement switch. However, in the Harmonics measurement the value is always reset to its default value on measurement switching, irrespective of the Retain Parameter setting.

In PXA, the following legacy parameters for the [:SENSe] :AVERAge :TYPE command are aliased:

Legacy Parameter	Aliased to	Action
LINear	SCALAr	Sets Scalar averaging
VOLTage	SCALAr	Sets Scalar averaging
VIDeo	LOG	Sets Log-Power averaging
LPOWer	LOG	Sets Log-Power averaging
POWer	RMS	Sets RMS averaging

In PXA, only the Swept SA measurement supports this functionality.

Detector 2 & 3 State

PXA command and query:

 :CALCulate:MAMarker:DETEctor[1]|2|3
OFF|NORMal|AVERage|POSitive|SAMPle|NEGative|QPEak|EAverage|RAverage

 :CALCulate:MAMarker:DETEctor[1]|2|3?

PSA commands, accepted by PXA for backwards compatibility:

 [:SENSe]:EMI:MEASure:DETEctor:QPEak[:STATE] OFF|ON|0|1

This command is supported for compatibility with the E7400 and PSA option 239. If sent with ON or 1 as parameter, it sets Detector 2 to QPEak. If sent with OFF or 0 as parameter, it sets Detector 2 to OFF.

 [:SENSe]:EMI:MEASure:DETEctor:AVERage[:STATE] OFF|ON|0|1

This command is included for compatibility with the E7400 and PSA option 239. If sent with ON or 1 as a parameter, it sets Detector 3 to EMI Average (EAverage). If sent with OFF or 0 as a parameter, it sets Detector 3 to OFF.

Swept SA Measurement Setup

Fixed / Relative Limit

Fixed / Relative Limit

PSA command and query, which are supported in PXA for backwards compatibility:

 :CALCulate:LLINE[1]|2:CMODE FIXED|RELative

 :CALCulate:LLINE[1]|2:CMODE?

PXA commands and queries

 :CALCulate:LLINE[1]|2|...|6:FREQuency:CMODE:RELative ON|OFF|1|0

 :CALCulate:LLINE[1]|2|...|6:FREQuency:CMODE:RELative?

 :CALCulate:LLINE[1]|2|...|6:AMPLitude:CMODE:RELative ON|OFF|1|0

 :CALCulate:LLINE[1]|2|...|6:AMPLitude:CMODE:RELative?

PSA offers only the **Limits Fixed / Rel** softkey, which is generic to all limit lines.

In PXA, this functionality is provided by a softkey that is specific to each limit line, and which provides a submenu with two softkeys (**Relative to CF** and **Relative to RL**). Programmatically, you can set relative amplitude and relative frequency independently for each limit line. :CALCulate:LLINE:CMODE RELative makes all limit lines relative to the center frequency and reference level.

In order to be consistent with the implementation of the PXA commands, the :CALCulate:LLINE:CMODE? query returns 1 if Limit Line 1 is set Relative to CF, and returns 0 otherwise.

In PSA, Limit Line functionality is available in every measurement and may be controlled via SCPI commands, even though the CCDF, Spectrum Emission Mask and Spurious Emissions measurements do not have softkey menus for Limit Line.

Limit Enabling

PSA and PXA command and query (valid suffixes 1-2 for PSA):

✓ :CALCulate:LLINE[1]|2|...|6:DISPlay OFF|ON|0|1

✓ :CALCulate:LLINE[1]|2|...|6:DISPlay?

PSA and PXA also accept the following command and query:

✓ :CALCulate:LLINE[1]|2:STATe OFF|ON|0|1

✓ :CALCulate:LLINE[1]|2:STATe?

In the past, you had to send the `DISPlay` command as well as the `STATe` command above in order to switch on a limit on and start testing. Now, the `DISPlay` command above is sufficient, but the `STATe` command is still accepted and is mapped to the `DISPlay` command.

In PSA, Limit Line functionality is available in every measurement and may be controlled via SCPI commands, even though the CCDF, Spectrum Emission Mask and Spurious Emissions measurements do not have softkey menus for Limit Line.

Swept SA Measurement Setup

Limit Line Data

Limit Line Data

PSA and PXA command and query (valid suffixes 1-2 for PSA):

 :CALCulate:LLINE[1]|2|...|6:DATA <x>,<ampl>,<connect>

 :CALCulate:LLINE[1]|2|...|6:DATA?

In the past it was possible to query the limit trace as though it were a normal trace, but in PXA, the query of the limit trace is not supported.

In PSA, Limit Line functionality is available in every measurement and may be controlled via SCPI commands, even though the CCDF, Spectrum Emission Mask and Spurious Emissions measurements do not have softkey menus for Limit Line.

Merge Limit Line Data

PSA and PXA command (valid suffixes 1-2 for PSA):



`:CALCulate:LLINE[1]|2|...|6:DATA:MERGe <x-axis>,<ampl>,<connected>`

In PSA, the limit is 200 points, but in PXA higher limits are accepted.

In PSA, Limit Line functionality is available in every measurement and may be controlled via SCPI commands, even though the CCDF, Spectrum Emission Mask and Spurious Emissions measurements do not have softkey menus for Limit Line.

Phase Noise Optimization

 [:SENSe]:FREQuency:SYNThesis[:STATe] 1|2|3

 [:SENSe]:FREQuency:SYNThesis[:STATe]?

 [:SENSe]:FREQuency:SYNThesis:AUTO[:STATe] OFF|ON|0|1

 [:SENSe]:FREQuency:SYNThesis:AUTO[:STATe]?

In PSA, this parameter is available in all measurements and is reset to measurement default on a measurement switch, because the Retain Parameter setting has no effect on this parameter.

In PXA, Phase Noise Optimization is supported only in certain measurements, as a measurement local parameter.

Note that, for PXA, the command forms above affect **only** the Swept SA measurement (see "[Measurement-Specific Commands](#)" on page 42 for more details). For other measurements, the command or query must include the measurement name, as follows:

 [:SENSe]:<meas>:FREQuency:SYNThesis[:STATe] 1|2|3

 [:SENSe]:<meas>:FREQuency:SYNThesis[:STATe]?

 [:SENSe]:<meas>:FREQuency:SYNThesis:AUTO[:STATe] OFF|ON|0|1

 [:SENSe]:<meas>:FREQuency:SYNThesis:AUTO[:STATe]?

The following measurements support these commands in PXA:

Measurement	<meas>
Adjacent Channel Power	ACPower
Channel Power	CHPower
List Sweep	LIST

4 PowerSuite Measurements Setup

The information in this chapter is divided into one general section that applies to multiple PowerSuite measurements, followed by sections that apply only to specific measurements.

For issues that relate to multiple measurements (or all measurements), see: “General” on page 42.

PowerSuite includes the measurements listed below. See the relevant section for information on issues specific to each measurement.

- “Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)” on page 45
- “Burst Power Measurement (BPOwer)” on page 51
- “Channel Power Measurement (CHPower)” on page 52
- “Complementary Cumulative Distribution Function Measurement (CCDF, PStatistic)” on page 53
- Harmonic Distortion Measurement (HARMonics). (No issues are listed for this measurement.)
- “Occupied Bandwidth Measurement (OBWidth)” on page 54
- “Spectrum Emission Mask Measurement (SEMAsk)” on page 55
- “Spurious Emissions Measurement (SPURious)” on page 57
- Third Order Intercept Measurement (TOI). (No issues are listed for this measurement.)

General

The information in this section applies to all PowerSuite measurements, or to more than one PowerSuite measurement, as specified.

Data Format for CONFigure Query

 :CONFigure?

PSA: The query return string includes double quotation marks, and uses the long-form name of the measurement.

PXA: The string does not include quotation marks, and returns the short-form name, as in the following example:

Instrument Type	Command	Response
PSA	CONF?	"CHPower"
PXA	CONF?	CHP

Measurement-Specific Commands

PSA: Commands in the groups listed below are applied to the currently-selected PowerSuite measurement.

PXA: The commands are always applied to the Swept SA measurement, regardless of which measurement is currently selected. To apply settings to the currently-selected measurement, you must specify the measurement name in the command, as described in the ["Example" on page 42](#) below.

- CALCulate:MARKer subsystem
- DISPlay:WINDow:TRACe subsystem
- TRACe:TYPE subsystem
- BANDwidth subsystem
- FREQuency subsystem
- SWEep subsystem
- DETector subsystem
- IF:GAIN
- TRIGger:SOURce subsystem

Example

Consider the case where the following query is sent to a PSA or a PXA with the ACPower measurement selected:

```
[ :SENSe ] :BANDwidth [ :RESolution ] ?
```

When this query is sent to a PSA, the value returned is the Resolution Bandwidth setting for the ACPower measurement.

When this query is sent to a PXA, the value returned is the Resolution Bandwidth setting for the *Swept SA* measurement. To obtain the Resolution Bandwidth setting for the ACPower measurement, you must send the query

```
[ :SENSe ] :ACPower :BANDwidth [ :RESolution ] ?.
```

Measurement Switching & Trigger Mode

 :CONFigure:ACPower|BPOWER|CHPower|EBWidth|OBWidth|PStatistic|SANalyzer|SEMask|SPURious

PSA: After switching to the specified measurement, the trigger mode is set to **Sweep Single**.

PXA: After switching to the specified measurement; the trigger mode is set to **Cont**.

Mode Catalog Query



INSTRUMENT:CATALOG?

- PSA returns only option names, while PXA returns the option name and option number (used by **INSTRUMENT:NSELECT**).
- PSA returns "DMODULATION" while PXA returns "VSA 100" for digital demodulation measurement option.

Mode Selection Commands



INSTRUMENT[:SELECT]

SA | SEQAN | EMI | BASIC | WCDMA | EDGE GSM | WIMAX OFDMA | VSA | PNOISE | NFIGURE | ADEMOD | BT | TDSCDMA | CDMA2K | CDMA1XEV | LTE | LTETDD | DVB | DTMB | DCATV | ISDBT | CMMB | CWLAN | CWIMAXOFDM | WIMAXFIXED | IDEN | RLC | SCPI LC | VSA89601



INSTRUMENT[:SELECT]?

PSA command, available in PXA for backwards-compatibility:



INSTRUMENT[:SELECT] GSM

When this command is sent to PXA, the parameter option GSM is mapped to **EDGE GSM**.

Mode Selection Behavior

In PSA, when the mode is changed, the instrument always switches to the default measurement for that mode.

Conversely, PXA remembers which measurement was last selected for each mode, and switches to that measurement whenever the mode is changed.

For example, if you had selected the ACP measurement in SA mode, then switched to GSM/EDGE mode, and then switched back to SA mode (using **INSTRUMENT:SELECT SA**), PSA would then be set to the Spectrum Analysis measurement, whereas PXA would be set back to the ACP measurement.

In PXA, to ensure a switch back to the SA measurement, you must explicitly follow the mode switch command with **CONFIGURE:SANALYZER**.

Radio Standard Profile

PSA command:



[:SENSE]:RADIO:STANDARD:PROFILE

PXA does not support this command, because there is no support for 802.11n.

Radio Standard Retain Parameters

PSA command and query:



[:SENSE]:RADIO:STANDARD:RPARAMS[:STATE] ON | OFF | 1 | 0



[:SENSE]:RADIO:STANDARD:RPARAMS[:STATE]?

PowerSuite Measurements Setup

General

In PSA, this command allows you to control whether radio standard parameters are retained after a measurement change.

In PXA, since parameters are automatically retained by each measurement, this command is not necessary, and is thus not supported.

Radio Standard Signal Band width

PSA command and query:

 [:SENSe]:RADio:STANdard:SBWidth <freq>

 [:SENSe]:RADio:STANdard:SBWidth?

In PSA, this command is used in conjunction with the Optimize Ref Level functionality (when the selected Radio Std is None).

PXA does not support the Optimize Ref Level functionality, so the command is not required and is not supported.

Restart Behavior

 :INITiate[:IMMediate]

 :INITiate:REStArt

For Spectrum Analysis mode in ESA and PSA, the **Restart** hardkey and the `INITiate:REStArt` command restart trace averages (and reset displayed average count to 1) for a trace in **Clear Write**, but do not restart **Max Hold** and **Min Hold**.

For wireless comms modes in ESA and PSA, the **Restart** hardkey and the `INITiate:REStArt` command restart every measurement, resetting all traces and numeric results.

In PXA, for **all** modes and Measurement Applications, the **Restart** hardkey and the `INITiate:REStArt` command reset not only **Trace Average**, but **Max Hold** and **Min Hold** traces as well.

Trigger Mode Switching Behavior

 :INITiate:CONTInuous OFF|ON|0|1

 :INITiate:CONTInuous?

For Spectrum Analysis mode in ESA and PSA, the trigger mode is controlled by a **Sweep Single/Cont** softkey. In these instruments, switching the **Sweep Single/Cont** softkey from Single to Cont restarts averages (and resets the displayed average count to 1), but does not restart **Max Hold** and **Min Hold**.

PXA has **Single** and **Cont** hardkeys, in place of the **Sweep Single/Cont** softkey. In PXA, if in single trigger mode, the **Cont** hardkey (and the `INITiate:CONInuous ON` command) switches to continuous mode, but never restarts a measurement and never resets a sweep.

Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)

In PXA, there is no separate Multi-Channel Power (MCPower) measurement. The functionality of the PSA MCPower measurement is subsumed under the Adjacent Channel Power measurement (ACPower). In most cases (but not all), MCPower-specific commands are aliased in PXA to ACPower commands.

Fetch, Measure & Read Commands for ACPower, MCPower

PSA queries:

 :READ:MCPower[n]?

 :FETCh:MCPower[n]?

 :MEASure:MCPower[n]?

PXA does **not** support these PSA queries. They are not aliased, so must be replaced with :READ:ACPower?, :FETCh:ACPower? and :MEASure:ACPower? respectively.

Note that, for $n = 1$, the PSA queries return 20 items. In PXA, the same queries may return a different number of items (depending on the measurement type), but the 20 PSA-compatible items are in the same order.

For the PSA query :FETCh:MCPower14?, the PXA query :FETCh:ACPower2? provides the same functionality.

Average Count & Average /Hold Number

PSA and PXA command and query:

 [:SENSe]:ACPower:AVERage[:STATe] OFF|ON|0|1

 [:SENSe]:ACPower:AVERage[:STATe]?

PSA command and query, aliased in PXA to the respective ACPower commands for backwards compatibility:

 [:SENSe]:MCPower:AVERage[:STATe] OFF|ON|0|1

 [:SENSe]:MCPower:AVERage[:STATe]?

In PXA this command operates the same way as in PSA, but has no effect on trace average.

The Trace Average Function is controlled by the [:SENSe]:AVERage[:STATe] command. When [:SENSe]:ACPower:AVERage[:STATe] is switched from on to off, [:SENSe]:AVERage[:STATe] is switched off.

Average Mode

PSA command and query, supported in PXA for backwards compatibility:

 [:SENSe]:MCPower:AVERage:TCONtrol EXponential|REPeat

 [:SENSe]:MCPower:AVERage:TCONtrol?

Carrier Integration Band width, Width & Power Present

PXA commands and queries:

PowerSuite Measurements Setup

Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:BANDwidth|BWIDth[:INTEgration] <freq>, <freq>, <freq>, <freq>, <freq>, <freq>`

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:BANDwidth|BWIDth[:INTEgration]?`

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:PPResent YES|NO, YES|NO, YES|NO, YES|NO, YES|NO, YES|NO`

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:PPResent?`

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:WIDTh <bandwidth>, <bandwidth>, <bandwidth>, <bandwidth>, <bandwidth>, <bandwidth>`

✓ `[:SENSe]:ACPower:CARRier[1]|2:LIST:WIDTh?`

PSA commands and queries, supported in PXA for backwards compatibility:

✓ `[:SENSe]:MCPower:CARRier:LIST:BANDwidth|BWIDth[:INTEgration] <Hz>`

✓ `[:SENSe]:MCPower:CARRier:LIST:BANDwidth|BWIDth[:INTEgration]?`

✓ `[:SENSe]:MCPower:CARRier:LIST:PPResent YES|NO`

✓ `[:SENSe]:MCPower:CARRier:LIST:PPResent?`

✓ `[:SENSe]:MCPower:CARRier:LIST:WIDTh <Hz>`

✓ `[:SENSe]:MCPower:CARRier:LIST:WIDTh?`

Initiate MCPower Measurement

✗ `:INITiate:MCPower`

PXA does **not** support this PSA command. It is not aliased, so must be replaced with `:INITiate:ACPower`.

Limit Test

PXA command and query:

✓ `:CALCulate:ACPower:LIMit:STATe OFF|ON|0|1`

✓ `:CALCulate:ACPower:LIMit:STATe?`

PSA commands and queries, supported in PXA for backwards compatibility:

`[:SENSe]:ACPower:LIMit[:STATe] OFF|ON|0|1`

✓ `[:SENSe]:ACPower:LIMit[:STATe]?`

✓ `[:SENSe]:MCPower:LIMit[:STATe] OFF|ON|0|1`

✓ `[:SENSe]:MCPower:LIMit[:STATe]?`

Meas Method

PXA command and query:

✔ [:SENSe]:ACPower:METHOD IBW|IBWRange|FAST|RBW

✔ [:SENSe]:ACPower:METHOD?

PSA command and query, supported in PXA for backwards compatibility:

✔ [:SENSe]:MCPower:METHOD IBW|RBW

✔ [:SENSe]:MCPower:METHOD?

Negative & Positive Offset Limits

PSA commands and queries for ACPower, supported in PXA for backwards compatibility:

⚠ :CALCulate:ACPower:OFFSET:LIST:LIMit:NEGative[:UPPer]:DATA <real>, <real>, <real>, <real>, <real>, <real>

⚠ :CALCulate:ACPower:OFFSET:LIST:LIMit:NEGative[:UPPer]:DATA?

⚠ :CALCulate:ACPower:OFFSET:LIST:LIMit:POSitive[:UPPer]:DATA <real>, <real>, <real>, <real>, <real>, <real>

⚠ :CALCulate:ACPower:OFFSET:LIST:LIMit:POSitive[:UPPer]:DATA?

These commands enable you to set the upper limit for the lower segment of the specified offset pair.

In PXA, the commands are implemented as aliases of [:SENSe]:ACPower:OFFSET[1]|2:LIST:RCARrier <real>, <real>, <real>, <real>, <real>.

However, the ACPower measurement in PXA does not support the functionality to specify limit values separately for positive and negative offsets. That is, the functionality of this command differs between PSA and MXA.

The following PSA MCPower commands and queries are **not** supported in PXA. These commands must be replaced in PXA with the ACPower equivalents.

✘ :CALCulate:MCPower:OFFSET:LIST:LIMit:NEGative[:UPPer]:DATA <dB>, <dB>, <dB>

✘ :CALCulate:MCPower:OFFSET:LIST:LIMit:NEGative[:UPPer]:DATA?

✘ :CALCulate:MCPower:OFFSET:LIST:LIMit:POSitive[:UPPer]:DATA <dB>, <dB>, <dB>

✘ :CALCulate:MCPower:OFFSET:LIST:LIMit:POSitive[:UPPer]:DATA?

Noise Correction State

✘ [:SENSe]:MCPower:CORRection:NOISe[:AUTO] OFF|ON|0|1

✘ [:SENSe]:MCPower:CORRection:NOISe[:AUTO]?

PowerSuite Measurements Setup

Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)

PXA does **not** support this PSA command and query. They are not aliased, so must be replaced with the [:SENSe]:ACPower:CORRection:NOISe[:AUTO] equivalents.

Number of Carriers

PXA command and query:

✔ [:SENSe]:ACPower:CARRier[1]|2:COUNT <integer>

✔ [:SENSe]:ACPower:CARRier[1]|2:COUNT?

PSA command and query, supported in PXA for backwards compatibility:

⚠ [:SENSe]:MCPower:CARRier:COUNT <integer>

⚠ [:SENSe]:MCPower:CARRier:COUNT?

In PSA, the default count value is 4, and the minimum value is 2. Note also that, on changing the number of carriers, the span is always recalculated.

In PXA, the default count value is 1, and the minimum value is 1.

Offset Filter Alpha & State

PSA and PXA command and query:

✔ [:SENSe]:ACPower:FILTer[:RRC]:ALPHa <real>

✔ [:SENSe]:ACPower:FILTer[:RRC][:STATe] OFF|ON|0|1

PSA command and query, aliased in PXA to the respective ACPower commands for backwards compatibility:

⚠ [:SENSe]:MCPower:FILTer[:RRC]:ALPHa <number>

⚠ [:SENSe]:MCPower:FILTer[:RRC][:STATe] OFF|ON|0|1

In PSA, this RRC Filter setting is applied to both carriers and offsets.

In PXA, this parameter is applied only to the offset, because the instrument can specify separate RRC filter settings for each carrier and offset.

Offset Frequency

PXA command and query:

✔ [:SENSe]:ACPower:OFFSet[1]|2:LIST[:FREQuency] <freq>, <freq>, <freq>, <freq>, <freq>, <freq>

✔ [:SENSe]:ACPower:OFFSet[1]|2:LIST[:FREQuency]?

PSA command and query, supported in PXA for backwards compatibility:

✔ [:SENSe]:MCPower:OFFSet:LIST[:FREQuency] <Hz>, <Hz>, <Hz>

✔ [:SENSe]:MCPower:OFFSet:LIST[:FREQuency]?

Offset Integration Bandwidth

PXA command and query:

✔ [:SENSe]:ACPower:OFFSet[1]|2:LIST:BANDwidth|BWIDth[:INTEgration] <bandwidth>, <bandwidth>, <bandwidth>, <bandwidth>, <bandwidth>

✔ [:SENSe]:ACPower:OFFSet[1]|2:LIST:BANDwidth|BWIDth[:INTEgration]?

PSA command and query, supported in PXA for backwards compatibility:

✔ [:SENSe]:MCPower:OFFSet:LIST:BANDwidth[:INTEgration] <Hz>, <Hz>, <Hz>

✔ [:SENSe]:MCPower:OFFSet:LIST:BANDwidth[:INTEgration]?

Reference Carrier

PXA commands and queries:

✔ [:SENSe]:ACPower:CARRier[1]|2:RCARRier <integer>

✔ [:SENSe]:ACPower:CARRier[1]|2:RCARRier?

✔ [:SENSe]:ACPower:CARRier[1]|2:RCARRier:AUTO OFF|ON|0|1

✔ [:SENSe]:ACPower:CARRier[1]|2:RCARRier:AUTO?

PSA commands and queries, supported in PXA for backwards compatibility:

✔ [:SENSe]:MCPower:RCARRier <integer>

✔ [:SENSe]:MCPower:RCARRier?

✔ [:SENSe]:MCPower:RCARRier:AUTO OFF|ON|0|1

✔ [:SENSe]:MCPower:RCARRier:AUTO?

Reference Carrier Frequency

PXA commands and queries:

✔ [:SENSe]:ACPower:CARRier[1]|2:RCFFrequency <freq>

✔ [:SENSe]:ACPower:CARRier[1]|2:RCFFrequency?

✔ [:SENSe]:ACPower:CARRier[1]|2:RCFFrequency:AUTO OFF|ON|0|1

✔ [:SENSe]:ACPower:CARRier[1]|2:RCFFrequency:AUTO?

PSA commands and queries, supported in PXA for backwards compatibility:

✔ [:SENSe]:MCPower:RCFFrequency <freq>

PowerSuite Measurements Setup

Adjacent Channel Power & Multi-Channel Power Measurements (ACPower, MCPower)

 [:SENSe]:MCPower:RCFRequency?

 [:SENSe]:MCPower:RCFRequency:AUTO OFF|ON|0|1

 [:SENSe]:MCPower:RCFRequency:AUTO?

Select Measurement

 :CONFigure:MCPower

PXA does **not** support this PSA command. It is not aliased, so must be replaced with :CONFigure:ACPower.

In PSA, the trigger mode is set to **Single** on any measurement change, as noted in "[Measurement Switching & Trigger Mode](#)" on page 42.

When using the CONFigure:ACPower command in PXA, as a replacement for PSA's CONFigure:MCPower, bear in mind that, in the PSA MCPower measurement, offsets A, B, C are **all** on, whereas in PXA ACPower measurement, **only** offset A is on. To enable offsets B and C in PXA, send ACPower:OFFSet:LIST:STATE 1,1,1.

Total Power Setting

PSA commands and queries, supported in PXA for backwards compatibility:

 [:SENSe]:MCPower:CARRier:AUTO[:STATe] OFF|ON|0|1

 [:SENSe]:MCPower:CARRier:AUTO[:STATe]?

 [:SENSe]:MCPower:CARRier[:POWer] <dBm>

 [:SENSe]:MCPower:CARRier[:POWer]?

Burst Power Measurement (BPOWer)

In PXA commands and queries, the keyword **BPOWer** is accepted as a synonym for **TXPower** in all cases, for backwards compatibility with PSA.

Fetch, Measure & Read Commands

 **:FETCh:TXPower|BPOWer[n]?**

 **:MEASure:TXPower|BPOWer[n]?**

 **:READ:TXPower|BPOWer[n]?**

Returned results differ slightly between PSA and PXA.

Average State

PXA and PSA command and query:

 **[:SENSe] :BPOWer|TXPower:AVERage[:STATe] OFF|ON|0|1**

 **[:SENSe] :BPOWer|TXPower:AVERage[:STATe]?**

In PXA this command operates the same way as in PSA, but has no effect on trace average.

The Trace Average Function is controlled by the **[:SENSe] :AVERage[:STATe]** command. When **[:SENSe] :ACPower:AVERage[:STATe]** is switched from on to off, **[:SENSe] :AVERage[:STATe]** is switched off.

Average Type

PXA command and query:

 **[:SENSe] :TXPower:AVERage:TYPE LOG|MAXimum|MINimum|RMS**

 **[:SENSe] :TXPower:AVERage:TYPE?**

PSA command and query, supported in PXA for backwards compatibility:

 **[:SENSe] :BPOWer:AVERage:TYPE LPOWer|POWer**

 **[:SENSe] :BPOWer:AVERage:TYPE?**

In PXA, the command **[:SENSe] :AVERage:TYPE** does not affect this setting, because its effect is specific to the Swept SA measurement.

Channel Power Measurement (CHPower)

Select Measurement



:CONFigure:CHPower

In PSA, the trigger mode is set to **Single** on any measurement change, as noted in “[Measurement Switching & Trigger Mode](#)” on page 42.

Also in PSA, switching to this measurement retains the Resolution Bandwidth settings from the SA measurement. However, if Resolution Bandwidth is set to MAN, and is greater than (SPAN * 0.01), Resolution Bandwidth and Video Bandwidth are changed to AUTO automatically. If Resolution Bandwidth is smaller than (SPAN * 0.002), Resolution Bandwidth is set to (SPAN * 0.0021).

Average State

PXA and PSA command and query:



[:SENSe] :CHPower :AVERAge [:STATe] OFF | ON | 0 | 1



[:SENSe] :CHPower :AVERAge [:STATe] ?

In PXA this command operates the same way as in PSA, but has no effect on trace average.

The Trace Average Function is controlled by the [:SENSe] :AVERAge [:STATe] command. When [:SENSe] :ACPower :AVERAge [:STATe] is switched from on to off, [:SENSe] :AVERAge [:STATe] is switched off.

Integration Bandwidth

PXA and PSA command:



[:SENSe] :CHPower :BANDwidth | BWIDth :INTEgration <bandwidth>

In PSA, this command also sets Span to the same as the Integration Bandwidth, whereas PXA sets Span to be 50% wider than Integration Bandwidth. This means that PXA must sweep a wider span, so could be slower than PSA.

Also in PSA, if Resolution Bandwidth is greater than (SPAN * 0.01), and Resolution Bandwidth is MAN, Resolution Bandwidth and Video Bandwidth are set to AUTO. If Resolution Bandwidth is narrower than (SPAN * 0.002), and Resolution Bandwidth is MAN, Resolution Bandwidth is set to (SPAN * 0.0021) and Video Bandwidth is set to AUTO.

Complementary Cumulative Distribution Function Measurement (CCDF, PStatistic)

Fetch, Measure & Read Commands

:FETCh:PStatistic[n]?

:MEASure:PStatistic[n]?

:READ:PStatistic[n]?

In PSA, sending the command with $n = 0$ returns the same results as for $n = 1$.

Conversely, in PXA, sending the command with $n = 0$ returns unprocessed I/Q data.

In PSA, sending the command with $n = 2, 3, \text{ or } 4$ returns a series of 501 floating point numbers. For PXA, the number of results returned for each of these cases is **5001**.

Occupied Bandwidth Measurement (OBWidth)

Fetch, Measure & Read Commands

PSA queries:

 :FETCh:OBWidth?, :MEASure:OBWidth?, :READ:OBWidth?

 :FETCh:OBWidth:OBWidth?, :MEASure:OBWidth:OBWidth?, :READ:OBWidth:OBWidth?

 :FETCh:OBWidth:FERRor?, :MEASure:OBWidth:FERRor?, :READ:OBWidth:FERRor?

 :FETCh:OBWidth:XDB?, :MEASure:OBWidth:XDB?, :READ:OBWidth:XDB?

PXA queries:

 :FETCh:OBWidth[n]?

 :MEASure:OBWidth[n]?

 :READ:OBWidth[n]?

In PSA, various result types for this measurement are obtained by adding an optional suffix keyword to the query, as listed above.

In PXA, the result types are obtained by adding an optional numeric suffix *n* (with value between 1 and 3) to the query.

Spectrum Emission Mask Measurement (SEMask)

Fetch, Measure & Read Commands

 :FETCh:SEMask[n]?

 :MEASure:SEMask[n]?

 :READ:SEMask[n]?

For certain values of n, the results returned by these queries differ between PSA and PXA, as detailed in the table below.

n	Comments
1	There are minor differences between the results definitions for PSA and PXA, when the Radio Standard is WLAN, or depending on the selected Meas Type.
2,3,4	In PSA, the number of frequency domain spectrum trace data points returned is determined by the setting of the trace points parameter (set by the command [:SENSe]:SPURious[:RANGe][:LIST]:SWEep:POINTs). In PXA, the number of frequency domain spectrum trace data points returned is 2001.
7,8	PSA: Returns 12 comma-separated pass/fail test results, determined by testing the absolute power in each offset against the specified offset's absolute power limits (14 when offset F is active). PXA: Returns 14 comma-separated pass/fail test results, determined by testing the minimum margin point from the limit line that is determined each offset's Limits setting.
12	In PSA, when the Radio Standard is WLAN, the instrument returns the band power result, otherwise it returns -999.0 In PXA, when Meas Type is Power Spectrum Peak Reference, the instrument returns the band power result (the peak power of the signal in the ref channel), otherwise it returns -999.0
13,14	In PXA, these values are supported only in the Multi Standard Radio (MSR) Measurement Application.

Average State

 [:SENSe]:SEMask:AVERage[:STATe]

In PSA, this parameter is completely coupled with [:SENSe]:AVERage[:STATe], whereas in PXA, the command without the SEMask keyword is specific to the Swept SA measurement.

Results Index

PSA command:

 [:SENSe]:SEMask:RESult

In PSA, this command allows you to view the sixth result when all offsets are active, because the display can show only five results at a time.

In PXA, all display lines are always visible, so this command is unnecessary and is **not** supported.

Sweep Time

PSA commands and queries, which are supported in PXA with the proviso described below:

PowerSuite Measurements Setup

Spectrum Emission Mask Measurement (SEMask)

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEeptime <time>... [<time>]

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEeptime?

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEeptime:AUTO OFF|ON|0|1

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEeptime:AUTO?

PXA commands and queries:

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEep[:TIME] <time>, ...

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEep[:TIME]?

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEep[:TIME]:AUTO ON|OFF|1|0, ...

 [:SENSe]:SEMask:OFFSet[1]|2:LIST:SWEep[:TIME]:AUTO?

Note that the **short** forms of `SWEeptime` and `SWEep` are the same. In PXA, for backwards compatibility, the `TIME` keyword is optional. Thus, the PSA commands and queries are compatible with PXA, provided that the short form `SWE` is used.

Spurious Emissions Measurement (SPURious)

Detector Settings

PSA commands and queries:

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor[:FUNction]
NORMal|AVERage|POSitive|SAMPle|NEGative|QPEak|EAverage|EPOSitive|MPositive

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor2[:FUNction]
OFF|NORMal|AVERage|POSitive|SAMPle|NEGative|QPEak|EAverage|EPOSitive|Mpositive

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor[2]?

PXA commands and queries:

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor[1][:FUNction]
AVERage|NEGative|NORMal|POSitive|SAMPle|RMS, ...

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor2[:FUNction]
OFF|AVERage|NEGative|NORMal|POSitive|SAMPle|RMS, ...

 [:SENSe]:SPURious[:RANGe][:LIST]:DETEctor[1]|2[:FUNction]?

The range of supported options differs between PSA and PXA. In PXA, `NORMal` is supported as a SCPI command parameter, but is aliased to `RMS`, so the query never returns `NORMal`.

Fast Spurious Measurement

PSA command and query, supported in PXA for backwards compatibility:

 [:SENSe]:SPURious:FSMeas OFF|ON|0|1

 [:SENSe]:SPURious:FSMeas?

PXA command and query, which provide the same functionality:

 [:SENSe]:SPURious:REPT:MODE ALL|LIMTest

 [:SENSe]:SPURious:REPT:MODE?

PowerSuite Measurements Setup
Spurious Emissions Measurement (SPURious)

5 Amplitude / Y Scale Commands

Attenuation Query



`[:SENSe] :POWer [:RF] :ATTenuation?`

The data format of the response to the query differs between PSA and PXA. PSA returns the value in floating point format, whereas PXA does not, as shown in the example below.

Instrument Type	Query	Response
PSA	POW:ATT?	+6.000E+01«x0A»
PXA	POW:ATT?	60«x0A»

Amplitude / Y Scale Commands Auto Scaling (Spurious Emissions Measurement)

Auto Scaling (Spurious Emissions Measurement)

PSA command and query, supported in PXA for backwards compatibility:

✔ `[:SENSe] :SPURious:POWer [:RF] :RANGe:AUTO OFF | ON | 0 | 1`

✔ `[:SENSe] :SPURious:POWer [:RF] :RANGe:AUTO?`

Equivalent PXA command and query:

✔ `:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y [:SCALE] :COUPle 0 | 1 | OFF | ON`

✔ `:DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y [:SCALE] :COUPle?`

Maximum Mixer Level

PSA and PXA command and query:

 [:SENSe]:POWER[:RF]:MIXer:RANGe[:UPPer] <real>

 [:SENSe]:POWER[:RF]:MIXer:RANGe[:UPPer]?

In PSA, all measurements are affected by this setting.

In PXA, only the Swept SA measurement supports this functionality, so this command has no effect on other measurements.

Amplitude / Y Scale Commands

Power Reference Value (Spurious Emissions Measurement)

Power Reference Value (Spurious Emissions Measurement)

PSA command and query, supported in PXA for backwards compatibility:

 [:SENSe]:SPURious:POWer[:RF]:LEVel <integer>

 [:SENSe]:SPURious:POWer[:RF]:LEVel?

In PSA, this parameter is completely coupled with :DISPlay:WINDow[1]:TRACe:Y[:SCALE]:RLEVel.

Equivalent PXA command and query:

 :DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALE]:RLEVel <real>

 :DISPlay:SPURious:VIEW[1]:WINDow[1]:TRACe:Y[:SCALE]:RLEVel?

Preselector Adjust

PXA command and query:

 [:SENSe]:POWer[:RF]:PADJust <freq>

 [:SENSe]:POWer[:RF]:PADJust?

The following PSA commands are accepted by PXA, and aliased to the command and query above respectively:

 [:SENSe]:POWer[:RF]:MW:PADJust

 [:SENSe]:POWer[:RF]:MMW:PADJust

PXA has only one preselector, so the PSA command and query below are accepted but the command has no effect:

 [:SENSe]:POWer[:RF]:PADJust:PRESelector MWAVE|MMWave|EXTernal

 [:SENSe]:POWer[:RF]:PADJust:PRESelector?

Amplitude / Y Scale Commands

Pre-Adjust for Min Clip

Pre-Adjust for Min Clip

PXA command and query:



```
[ :SENSe ] :POWER [ :RF ] :RANGe :OPTimize :ATTenuation OFF | ELECtrical | COMBined
```



```
[ :SENSe ] :POWER [ :RF ] :RANGe :OPTimize :ATTenuation?
```

PSA command and query:



```
[ :SENSe ] :POWER [ :RF ] :RANGe :AUTO ON | OFF | 1 | 0
```



```
[ :SENSe ] :POWER [ :RF ] :RANGe :AUTO?
```

PXA accepts the PSA command and query, and aliases them respectively. PXA aliases command parameters as follows:

ON	ELECtrical
OFF	OFF

PXA returns `TRUE` to the PSA query, if the current setting is not `OFF`.

Reference Level

PSA command. In PXA, this command applies only to the Swept SA measurement.

 :DISPlay:WINDow[1]:TRACe:Y[:SCALE]:RLEVel <real>

 :DISPlay:WINDow[1]:TRACe:Y[:SCALE]:RLEVel?

Note that in PSA, this setting is shared between Spectrum Analyzer and all PowerSuite measurements (except CCDF), whereas in PXA the setting is stored per measurement.

Therefore, in PXA, for measurements other than Swept SA, you must use commands of the form:

 :DISPlay:<meas>:VIEW[1]:WINDow[1]:TRACe:Y[:SCALE]:RLEVel <real>

If you send a larger RLEVel value than the maximum allowed, PXA generates the error message “-222, Data Out of Range”, whereas PSA does not generate an error, and rounds the value without notification. If a program written for PSA checks the error queue, this may cause a problem when the code is applied to PXA.

In PSA, the maximum value of Reference Level is 0 dBm when the preamp is on. This restriction does not apply to PXA.

Reference Level is a display function, not a measurement control function, so in PXA a change in the setting does not start a new sweep (unless attenuation changes).

Reference Level Offset



:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_amp1>



:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?

In PSA, the Reference Level Offset cannot be adjusted using the knob or step keys. That is no longer the case in PXA.

In PSA, this setting applies to all measurements, including CCDF (even though CCDF does not have corresponding soft panel controls).

In ESA and PSA, Reference Level Offset is applied to the data as it is acquired, thus if the offset changes, the new offset value is not applied until new trace data is obtained. In PXA, the offset is applied as the data is displayed or queried, so if you change the offset, the data changes immediately.

Scale/Div

PSA command. In PXA, this command applies only to the Swept SA measurement.



`:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:PDIVision <rel_amp1>`

Note that in PSA, this setting is shared between Spectrum Analyzer and all PowerSuite measurements (except CCDF), whereas in PXA the setting is stored per measurement.

Therefore, in PXA, for measurements other than Swept SA, you must use commands of the form:



`:DISPlay:<meas>:VIEW[1]:WINDow[1]:TRACe:Y[:SCALE]:PDIVision <rel_amp1>`

If you send a larger `PDIVision` value than the maximum allowed, PXA generates the error message “-222, Data Out of Range”, whereas PSA does not generate an error, and rounds the value without notification. If a program written for PSA checks the error queue, this may cause a problem when the code is applied to PXA.

Amplitude / Y Scale Commands

Scale Type

Scale Type



`:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:SPACing LINear|LOGarithmic`

In PSA, this functionality is only available in the SA and Third Order Intercept (TOI) measurements.

In PXA, only the Swept SA measurement supports this functionality.

Y Axis Unit

 :UNIT:POWer

In PSA, this command affects all measurements, except as follows:

- In the Spectrum Emission Mask (SEM) and Spurious Emissions (SPUR) measurements, this command does not affect Trace Graph, but the result of power calculation is affected.
- The Complementary Cumulative Distribution Function (CCDF) measurement does not support this functionality.

In PXA, only the Swept SA measurement supports this functionality.

Amplitude / Y Scale Commands
Y Axis Unit

6 Bandwidth Commands

In PSA, the relevant hardkey is labeled **BW/Avg** and includes menu keys to control the averaging behavior of the instrument, which is global.

In PXA, the hardkey is labeled **BW**, and averaging is performed on a trace-by-trace basis, with a corresponding impact on the SCPI functions. A backwards compatibility command (`[:SENSe] :AVERAge [:STATe]`) is provided to preserve the previous global behavior (see [“Trace Mode \(Type\)” on page 138](#)).

Note that in PXA, the Average functions, which appear in the **BW** menu in PSA, can be found in the Trace/Detector menu and the Meas Setup menu. In the Trace/Detector menu, you may turn Trace Averaging on or off for the desired traces (rather than globally as in the past); in the Meas Setup menu you may configure Averaging, by setting the Average Number and the Average Type.

EMC Standard

 `[:SENSe] :BANDwidth | BWIDth : [:RESolution] :MODE EMI | SAN | OFF`

 `[:SENSe] :BANDwidth | BWIDth : [:RESolution] :MODE?`

EMC Standard (for Backwards Compatibility Only)

Bandwidth Commands

Resolution Bandwidth

Resolution Bandwidth

 [:SENSe]:BANDwidth|BWIDTH[:RESolution] <freq>

 [:SENSe]:BANDwidth|BWIDTH[:RESolution]?

 [:SENSe]:BANDwidth|BWIDTH[:RESolution]:AUTO OFF|ON|0|1

 [:SENSe]:BANDwidth|BWIDTH[:RESolution]:AUTO?

In PSA, these settings are affected by Retain Parameter. When Retain Parameter is on, values are retained between measurement switches. When Retain Parameter is off, values are reset to their defaults.

In the Harmonics, Spectrum Emission Mask (SEM) and Spurious Emissions (SPUR) measurements, the frequency value is overridden by the value defined in Range Table.

In PXA, these settings are always retained as Meas Local parameters.

Video Band width

 [:SENSe]:BANDwidth|BWIDth:VIDeo <freq>

 [:SENSe]:BANDwidth|BWIDth:VIDeo?

 [:SENSe]:BANDwidth|BWIDth:VIDeo:AUTO OFF|ON|0|1

 [:SENSe]:BANDwidth|BWIDth:VIDeo:AUTO?

In PSA, these settings are affected by Retain Parameter. When Retain Parameter is on, the values are retained between measurement switches. When Retain Parameter is off, the values are reset to their defaults.

In the Harmonics and Spurious Emissions (SPUR) measurements, the frequency value is overridden by the value defined in Range Table.

In the Spectrum Emission Mask (SEM) measurement, although it is expected that Video Bandwidth will be calculated from the Resolution Bandwidth, the **AUTO** setting may be changed to **OFF**, and it is retained if Retain Parameter is on.

In PXA, these settings are always retained as Meas Local parameters.

Bandwidth Commands
Video Bandwidth

7 Display Commands

Display Enable

 :DISPlay:ENABle OFF|ON|0|1

 :DISPlay:ENABle?

In PXA, :SYSTem:PRESet no longer enables the display.

Display Commands

Full Screen Display

Full Screen Display

 :DISPlay:FSCReen[:STATe] OFF|ON|0|1

 :DISPlay:FSCReen[:STATe]?

In PXA, the following alias is provided for ESA backwards compatibility:

 :DISPlay:MENU[:STATe] OFF|ON|0|1

In PSA, Full Screen is switched on with a softkey, so pressing any other key turns Full Screen off.

In PXA, because a hardkey is provided to turn this function on and off, pressing any other key no longer switches off Full Screen.

Window Annotation

PXA/PSA command and query:

 :DISPlay:WINDow[1]:ANNotation[:ALL] OFF|ON|0|1

 :DISPlay:WINDow[1]:ANNotation[:ALL]?

In PXA, the `WINDow` parameter and optional subopcode are included for backwards compatibility, but ignored. The command affects **all** windows equally.

The Complementary Cumulative Distribution Function (CCDF) measurement does not support this functionality.

Display Commands

Display Line Amplitude & State

Display Line Amplitude & State



`:DISPlay:WINDow[1]:TRACe:Y:DLINe <amp;gt;`



`:DISPlay:WINDow[1]:TRACe:Y:DLINe:STATe OFF|ON|0|1`

In PXA, only the Swept SA measurement supports this functionality.

8 Frequency Commands

Center Freq



[:SENSe] :FREQuency:CENTer <freq>

In PSA, if in the Channel Power measurement you change the center frequency while the Video Bandwidth setting is MAN, Video Bandwidth is set to AUTO once and then reverts to MAN. PXA does not do this, which could create a situation where Video Bandwidth settings unintentionally differ between PSA and PXA.

PXA does **not** retain the marker display position. To emulate the PSA marker behavior in PXA, use the following algorithm.

1. Acquire the current marker position (X1)
2. Acquire the current Center and Span.
3. Calculate the display position Ratio (-50% to +50%) of the marker:
Ratio = (X1 - Center) / Span
4. Change the center frequency (Center2)
5. Move marker to the new position, which is given by: Center2 + Ratio * Span

Frequency Commands

Start & Stop Frequencies

Start & Stop Frequencies

These PSA commands and queries are supported in PXA, but **only** for the Swept SA measurement:

 [:SENSe]:FREQuency:START <freq>

 [:SENSe]:FREQuency:START?

 [:SENSe]:FREQuency:STOP <freq>

 [:SENSe]:FREQuency:STOP?

In PXA, for measurements other than Swept SA, use the following sequence instead to set the displayed frequency range:

 [:SENSe]:FREQuency:CENTer <freq>

 [:SENSe]:<meas>:FREQuency:SPAN <freq>

In PSA, setting a new frequency does **not** move the marker display position. In PXA, markers remain at the same frequency, so the display position moves. Therefore, to retain the marker display position in PXA (as for PSA), you must move the marker after you set a new frequency.

Frequency Offset

 `[:SENSe] :FREQuency:OFFSet <freq>`

 `[:SENSe] :FREQuency:OFFSet?`

In PXA, the command `:DISPlay:WINDow[1]:TRACe:X[:SCALe]:OFFSet` is supported for backwards compatibility, but is not recommended for new development.

In PSA, Frequency Offset cannot be adjusted by the knob or step keys. That is no longer the case in PXA.

The following PSA command is accepted by PXA, and aliased to the command above:

 `:DISPlay:WINDow[1]:TRACe:X[:SCALe]:OFFSet <freq>`

Frequency Span



[:SENSe] :FREQuency:SPAN <freq>



[:SENSe] :FREQuency:SPAN?

Marker Display Position: Just as for [:SENSe] :FREQuency:START and [:SENSe] :FREQuency:STOP, PSA retains its marker display position after changing span. PXA does not retain the marker position, so you must move the marker after changing span. For details of how to do this, see ["Start & Stop Frequencies" on page 80](#).

Band Power Span Setting: PSA retains its Band Power band span after Span is changed (because band is based on delta markers). PXA does **not** retain this setting, so you must move the band start and stop positions after changing span, using :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:LEFT and :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:RIGHT.

Marker to Center: The following command sequence sets the marker to the center in PSA, but not in PXA. PXA changes to zero span before the marker is moved with [:SENSe] :FREQuency:MARKer:CENTer. To avoid this, insert *OPC? after :CALCulate:MARKer:CENTer in the sequence.

```
:CALCulate:MARKer:MAXimum
```

```
:CALCulate:MARKer:CENTer
```

```
[ :SENSe ] :FREQuency:SPAN 0
```

Zero Span and Marker Position: Usually setting to zero span does not change the marker display position in PXA. However, when trace is not updated after switching measurement with :CONFIgure:SANalyzer:NDEFault, setting zero span moves the marker to the left edge of the span. This does not happen in PSA.

Measurement-Specificity: In PXA, for measurements other than Swept SA, use the command [:SENSe] :<meas>:FREQuency:SPAN instead of [:SENSe] :FREQuency:SPAN.

In PSA, for the Complementary Cumulative Distribution Function (CCDF), Harmonics, and Burst Power measurements, the query form [:SENSe] :FREQuency:SPAN? returns 0. For these measurements, PXA does **not** support the equivalent [:SENSe] :<meas>:FREQuency:SPAN? and does not return anything.

Integration Band width: PSA's Channel Power measurement allows you to set a narrower span than the Integration Bandwidth. PXA does **not** allow this and generates an error. Thus, code written for PSA that sets Span first and Integration Bandwidth later may cause problems in PXA, even though it works correctly in PSA.

Span/Resolution Bandwidth Ratio

 [:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio <integer>

 [:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?

 [:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO OFF|ON|0|1

 [:SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO?

In PXA, only the Swept SA measurement supports these parameters.

Frequency Commands Full Span & Last Span

Full Span & Last Span



[:SENSe] : FREQuency : SPAN : FULL

[:SENSe] : FREQuency : SPAN : PREVIous

In PSA, this functionality is available in the measurement which Span is available.

In PXA, this functionality is supported as a measurement local parameter.

In PSA, the **Full Span** function turns off all markers. In PXA, this is not the case.

Global Center Frequency

PSA and PXA command and query:

✔ :GLOBal:FREQuency:CENTer[:STATe] 1|0|ON|OFF

✔ :GLOBal:FREQuency:CENTer[:STATe]?

In PXA, the command and query are aliased respectively to:

✔ :INSTrument:COUPle:FREQuency:CENTer ALL|NONE

✔ :INSTrument:COUPle:FREQuency:CENTer?

Frequency Commands
Global Center Frequency

9 Input/Output Commands

Most of the settings in the PXA Input/Output system, including External Gain, Amplitude Corrections settings and data, etc., are shared by all modes and are not changed by a mode switch. Furthermore, most variables in the Input/Output system key are not affected by Mode Preset. Both of these behaviors represent a departure from legacy behavior.

In PXA, Input/Output settings are reset by using the "Restore Input/Output Defaults" function. They can also be reset to their default values by the **System > Restore System Defaults > In/Out Config** key or through the **System > Restore System Defaults > All** key (or equivalent SCPI commands).

While this matches most use cases better, it does create some code compatibility issues. For example, Amplitude Corrections are no longer turned off by a Mode Preset, but instead by using the "Restore Input/Output Defaults" key or SCPI command.

Although Input/Output settings are not part of each Mode's State, they are saved in the Save State files, so that all of the instrument settings can be recalled with Recall State, as in legacy instruments.

Input/Output Commands

Analog Out

Analog Out

PXA commands and queries:

✓ :OUTPut:ANALog OFF|SVIDeo|LOGVideO|LINVideO|DAUDio

✓ :OUTPut:ANALog:AUTO OFF|ON|0|1

✓ :OUTPut:ANALog:AUTO?

PSA command:

✓ :OUTPut:ANALog SANalyzer|DNWB

PSA and PXA query:

✓ :OUTPut:ANALog?

In PXA, prior to A.04.00, OFF was the default setting, except when in the Analog Demod application or with Tune and Listen, in which case it was DAUDio, and there was no selection menu. Thus, for backwards compatibility with earlier PXA firmware versions, AUTO (:OUTPut:ANALog:AUTO ON) duplicates the prior behavior.

The DNWB and SANalyzer parameters, which are valid in PSA but perform no function in PXA, are accepted by PXA without error.

Aux IF Out

 :OUTPut :AUX SIF|AIF|LOGVideo|OFF

 :OUTPut :AUX?

In PSA, the IF output has functionality equivalent to the "Second IF" function in PXA's Aux IF Out menu. In PXA, it is necessary to switch the Aux IF Out to "Second IF" to obtain this functionality, whereas in PSA it is always on, since there are no other choices.

Hence, when code written for PSA that uses the IF Output is ported to PXA, a command to switch this function to "Second IF" must be added.

Correction Set Commands

The following commands and queries are valid in PSA and PXA, except that for PSA there only four correction sets ([1] | 2 | 3 | 4), rather than six.

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6[:STATe] ON|OFF|1|0

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6[:STATe]?

In PXA, Preset does not switch Corrections off. Restore Input/Output Defaults does switch off Corrections.

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:DELeTe

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:X:SPACing LINear|LOGarithmic

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:X:SPACing?

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:DATA <freq>, <ampl>, ...

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:DATA?

 [:SENSe]:CORRection:CSET[1] | 2 | ... | 6:DATA:MERGe <freq>, <ampl>, ...

 [:SENSe]:CORRection:CSET:ALL:DELeTe

 [:SENSe]:CORRection:CSET:ALL[:STATe] ON|OFF|1|0

 [:SENSe]:CORRection:CSET:ALL[:STATe]?

In PSA, these commands affect all measurements including CCDF, although CCDF does not have a soft menu for the functionality.

External Preamp Gain

PXA command and query:

 [:SENSe]:CORRection:SA[:RF]:GAIN <rel_amp>

 [:SENSe]:CORRection:SA[:RF]:GAIN?

PSA command and query also supported by PXA for backwards compatibility:

 [:SENSe]:CORRection:OFFSet[:MAGNitude] <relative_power>

 [:SENSe]:CORRection:OFFSet[:MAGNitude]?

The legacy "Ext Preamp Gain" key is now called "Ext Gain" and has menu choices: Ext Preamp, MS, and BTS, for backwards compatibility.

RF Calibrator Signal Selection

PXA command and query:

 [:SENSe]:FEED RF|AIQ|EMIXer

 [:SENSe]:FEED?

PSA command and query also supported by PXA for backwards compatibility:

 [:SENSe]:FEED AREFERENCE

In PSA the calibrator is one of the inputs and is selected using the **AREFERENCE** parameter of the same **:FEED** command that switches the inputs. In PXA, it is controlled in a separate menu, and overrides the input selection. For code compatibility the [:SENSe]:FEED AREFERENCE command is provided, and is aliased to [SENSe]:FEED:AREFERENCE REF50, which causes the input to be switched to the 50 MHz calibrator. The [:SENSe]:FEED RF command switches the input back to the RF port and turns the calibrator off, thus providing full compatibility with the PSA calibrator function.

Note that after sending this command, the query [:SENSe]:FEED? does **not** return **AREF** but instead the currently selected input.

Input Mixer

PSA command and query:

 :INPut:MIXer EXTErnal|INTErnal

 :INPut:MIXer?

In PSA, you choose between the Internal mixer or an External Mixer, whereas in PXA, the External Mixer is one of the choices for the input and hence is selected using the FEED command (:SENSe:FEED EMIXer).

In PXA, the :INPut:MIXer EXTErnal|INTErnal command and query are mapped as follows:

- When :INPut:MIXer EXTErnal is received, :SENSe:FEED EMIXer is executed.
- When :INPut:MIXer INTErnal is received, :SENSe:FEED RF is executed.
- When :INPut:MIXer? is received, the response will be **INT** if any input other than the external mixer is selected, or **EXT** if the external mixer is selected.

PSA command and query:

 :INPut:MIXer:TYPE PRESelected|UNPreselect

 :INPut:MIXer:TYPE?

PXA does **not** support the :INPut:MIXer:TYPE command or query.

10 Marker Commands

General Differences

Position Markers vs. Value Markers

In PSA, markers are position markers, which means that Normal and Delta markers stay at the same screen position when X Axis parameters are changed.

In PXA, markers are value markers, which means that when the instrument's X Axis settings are changed, the marker's X Axis value in fundamental X Axis units remains unchanged.

As a result of this change, markers in PXA may be set to a frequency that is offscreen, whereas in PSA, they are always clipped to the screen edges and hence are never offscreen.

In PSA, querying a marker's position always return a valid result, because a marker can never be offscreen. In PXA, querying the position of a marker that happens to be offscreen returns Not A Number, which can never happen in PSA.

Band Power Functionality & Marker Modes

In PXA, the introduction of adjustable-width Band Functions fundamentally changes the way Band Power markers are controlled, by using a single marker to completely define the function.

To define the Band Power function, PSA uses Delta Marker functionality with two markers, for example, Marker 1 and its Reference Marker.

The PSA Band Power marker modes, known as Span Pair and Delta Pair (Band Pair in ESA), are used to set two markers for the primary purpose of defining the band of a Band Power function. The two markers are set by adjusting their span and centerpoint (Span Pair mode) or by adjusting their locations independently to directly define the Start and Stop edges of the band (Band Pair/Delta Pair modes).

To set up Band Power measurements in PSA, you send the `:CALCulate:MARKer[1]|2|3|4:MODE` command with either the `BAND` or `SPAN` parameter, in order to turn on those marker control modes. In PXA, this is no longer necessary, as there are no special marker modes for Band Power. The `BAND` and `SPAN` parameters are accepted by PXA but aliased, as described in ["Marker Type" on page 95](#).

In PXA the marker itself has a width attribute, which you set using the Band Span function. The marker shows "wings" that define the edges of the band in which the Band Power is being measured. You only need one marker, not a pair of markers, to completely define a Band Power function. Hence in PXA it is possible to do Delta Band Power, which PSA and ESA cannot do.

Additional control functions Band Left and Band Right are provided for the case when you need to precisely set the band edges. Note that the marker itself always remains centered in the band.

Marker Commands

General Differences

Aliases and compatibility commands are available in PXA to map the PSA Span Pair and Band Pair/Delta Pair functions for code compatibility, as described in [“Span Pair Commands” on page 98](#), [“Delta Pair \(Band Pair\) Commands” on page 99](#) and [“Marker Function Settings” on page 119](#). Since Span Pair and Band Pair/Delta Pair are primarily used for making band power measurements, the aliases are provided for setting the parameters of a Band Function. Note that, if you use the PSA commands for anything other than Band Power, these aliases are not likely to yield compatible results.

For example, the Band Pair commands let you arbitrarily set the frequency (time) of a delta marker and its non-fixed reference marker. In these cases, the aliased commands will not be compatible. For these cases, you must use two markers and position each using the `CALCulate:MARKer[1]|2|. . . |12:X` command, since “marker pairs” do not exist in PXA.

Band Power Settings & Marker Screen Positions

In PSA, when a marker pair is used to set the width of the band for Band Power, the markers hold their screen positions when other frequency settings (such as Span) change. The result of this is that as the Span changes, the frequency difference and hence the width of the band change as well.

In PXA, as a result of the change from position markers to value markers, the width of the band remains constant as frequency settings of the instrument change.

Marker Commands & PowerSuite Measurements

In PSA, the CCDF, Spectrum Emission Mask and Spurious Emissions measurements have their own local marker systems, so most marker commands that lack the `<meas>` keyword do not affect those measurements.

Also in PSA, markers are switched off on a measurement switch, even if the Retain Parameter setting is on.

In PXA, commands without a `<meas>` keyword affect only the Swept SA measurement.

All Markers Off

PSA and PXA command:

 :CALCulate:MARKer:AOff

For applicability of these commands to the selected measurement, see “[Marker Commands & PowerSuite Measurements](#)” on page 94. Note that, in PXA, the form of the command without a <meas> keyword (as shown here) applies **only** to the Swept SA measurement.

Marker Type

For applicability of these commands to the selected measurement, see “[Marker Commands & PowerSuite Measurements](#)” on page 94.

 :CALCulate:MARKer[1]|2|...|12:MODE POSition|DELTA|FIXed|OFF

 :CALCulate:MARKer[1]|2|...|12:MODE?

PSA command and query:

 :CALCulate:MARKer[1]|2|3|4:MODE POSition|DELTA|BAND|SPAN|OFF

 :CALCulate:MARKer[1]|2|3|4:MODE?

In PXA, for compatibility with PSA's Band Power functionality, the **BAND** and **SPAN** parameters are accepted, and both are aliased to **POSITION**. See “[Band Power Functionality & Marker Modes](#)” on page 93 for more information. The PXA query never returns **BAND** or **SPAN**.

In PSA, selecting the **DELTA** option always moves the reference marker to the delta marker. In PXA, it only does so if the marker was already a delta marker.

In PSA, only a Reference marker may be **FIXed**: it is always **FIXed** and cannot be moved. In PXA, **any** marker can be set to **FIXed**, and may be moved to any X or Y value.

In PSA, sending the command with the **POSITION** option when Band Power functionality is enabled switches off Band Power measurement. This is not the case for PXA, where you have to explicitly switch off Band Power measurement, by sending :CALCulate:MARKer:FUNCTION OFF.

Similarly, PSA does **not** switch off Noise Marker on receipt of :CALCulate:MARKer:MODE POSITION.

PSA command and query, supported in PXA for backwards compatibility:

 :CALCulate:MARKer[1]|2|...|12:STATE OFF|ON|0|1

 :CALCulate:MARKer[1]|2|...|12:STATE?

Marker X Axis Value

 :CALCulate:MARKer[1]|2|...|12:X <freq|time>

 :CALCulate:MARKer[1]|2|...|12:X?

See [“Position Markers vs. Value Markers”](#) on page 93.

In PSA, a marker always remains on-screen. Setting X to a value outside the span rounds the value to the right or left edge of the display. Programs written for PSA may use this behavior to move a marker to the display edges, by setting an arbitrary extremely small or large value.

In PXA, markers are frequency-based, and values are never rounded to fit within the display. You can move a marker to any specified frequency, regardless of whether it is within the displayed span.

If a marker is set to an extremely small or large value in PXA, no measurement will be made if, as a result, the marker is outside the span.

To move a marker to the start or stop frequency in PXA, query the start or stop frequency, then set the exact frequency with the `CALCulate:MARKer[1]|2|...|12:X` command.

Note that you cannot move a marker to the start frequency by using `CALCulate:MARKer:X:POSITION 0`. See [“Marker X Axis Position \(in Trace Points\)”](#) on page 97 for details.

Marker X Axis Position (in Trace Points)

 :CALCulate:MARKer[1]|2|...|12:X:POSition <real>

 :CALCulate:MARKer[1]|2|...|12:X:POSition?

As described in “Delta Pair (Band Pair) Commands” on page 99, some PSA code uses that instrument’s rounding behavior to place markers at edge-of-screen start or stop frequencies, by specifying off-screen negative or positive position values.

This technique does **not** work in PXA for 2 reasons:

1. PXA does not round values to maximum or minimum trace points. If you specify a value outside the screen edges, the marker simply moves to a position beyond the display.
2. PXA’s “trace point” is tied to trace, not to display. When you change the center or span frequency, the trace point table is not updated until you initiate a sweep. This means the trace point and display point are not synchronized, so sending :CALCulate:MARKer:X:POSition 0 does not move the marker to the left edge of the display. To avoid this problem, perform a sweep, or use :CALCulate:MARKer:X instead.

Marker Commands

Span Pair Commands

Span Pair Commands

In PSA, the Span Pair function is used with a marker pair to set the band for Band Power. The following commands and queries are available for performing this setup:

 :CALCulate:MARKer[1]|2|3|4:X:CENTer <param>

 :CALCulate:MARKer[1]|2|3|4:X:CENTer?

 :CALCulate:MARKer[1]|2|3|4:X:SPAN <param>

 :CALCulate:MARKer[1]|2|3|4:X:SPAN?

In PXA, the Band Power functionality has been superseded by adjustable-width Band Functions (see “[Band Power Functionality & Marker Modes](#)” on page 93 for details). Hence, the PSA commands are aliased as follows to preserve the PSA Band Power functionality as much as possible:

PSA command/query	Aliased to in PXA
:CALCulate:MARKer[1] 2 3 4:X:CENTer	:CALCulate:MARKer[1] 2 3 4:X
:CALCulate:MARKer[1] 2 3 4:X:SPAN	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:SPAN

Delta Pair (Band Pair) Commands

In PSA, the Delta Pair function (Band Pair in ESA) may also be used to set the marker pair for Band Power. The following commands and queries are available for performing this setup:

 :CALCulate:MARKer[1]|2|3|4:X:START <param>

 :CALCulate:MARKer[1]|2|3|4:X:START?

 :CALCulate:MARKer[1]|2|3|4:X:STOP <param>

 :CALCulate:MARKer[1]|2|3|4:X:STOP?

In PXA, the Band Power functionality has been superseded by adjustable-width Band Functions (see “[Band Power Functionality & Marker Modes](#)” on page 93 for details). Hence, the PSA commands are aliased as follows to preserve the PSA Band Power functionality as much as possible:

PSA command/query	Aliased to in PXA
:CALCulate:MARKer[1] 2 3 4:X:START	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:LEFT
:CALCulate:MARKer[1] 2 3 4:X:STOP	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:RIGHT

(In PXA, there are 12 available marker indexes for all these commands, although the examples above show the 4 indexes available in PSA.)

Note that the value of any marker in PXA may be set to a frequency that is offscreen, whereas in PSA, values are always clipped to the screen edges, and hence markers are never offscreen. If PSA code depends on this clipping behavior, by setting Band Span to extreme values in order to force Band Power markers to the left and right edges of the screen, it is unlikely to work correctly in PXA.

Arbitrary Marker Pair functionality

In PSA, it is also possible to use the `START` and `STOP` commands to arbitrarily set the frequency (or time) of a delta marker and its reference marker, without being in Band Power mode. This case is **not** supported by any backwards compatibility commands in PXA. However, since in PXA you can arbitrarily set any marker’s value and any reference marker’s value, it is possible to modify PSA code to emulate this behavior, as shown in the example below:

Example PSA command:	Change to PXA command:
:CALCulate:MARKer1:X:START <freq1>	:CALCulate:MARKer1:X <freq1>
:CALCulate:MARKer1:X:STOP <freq2>	:CALCulate:MARKer2:X <freq2>

In the example above, Marker 1 and Marker 2 are used; in practice, you would use the reference marker number for the `STOP` marker number, which is usually [marker number + 1].

Marker Commands

Band Interval Left (Delta Pair Start Position)

Band Interval Left (Delta Pair Start Position)

PSA command and query, supported in PXA for backwards compatibility:

 :CALCulate:MARKer[1]|2|3|4:X:POSition:START <integer>

 :CALCulate:MARKer[1]|2|3|4:X:POSition:START?

PXA command and query:

 :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:LEFT <freq>

 :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:LEFT?

In PSA, the command `:CALCulate:MARKer[n]:X:POSition:START <param>` is used to control the Reference marker in trace points (buckets) in Band Pair/Delta Pair mode. PXA supports this command for backwards compatibility by first converting the specified value in trace points to the current X Axis Scale Units (frequency or time) of the trace upon which the marker resides. Then, that value is used in the `:CALC:MARKer[n]:FUNction:BAND:LEFT <param>` command to set the start of the marker's Band Function.

Note also that PXA's "trace point" is tied to trace, not to display. When you change the center or span frequency, the trace point table is not updated until you initiate a sweep. This means the trace point and display point are not synchronized, so sending `:CALCulate:MARKer:X:POSition 0` does not move the marker to the left edge of the display. To avoid this problem, perform a sweep, or use `:CALCulate:MARKer:X` instead.

The query returns the marker function `LEFT` value in trace points (buckets), by translating the value based on the current X Axis Scale settings.

Band/Interval Right (Delta Pair Stop Position)

PSA command and query, supported in PXA for backwards compatibility:

 :CALCulate:MARKer[1]|2|3|4:X:POSition:STOP <integer>

 :CALCulate:MARKer[1]|2|3|4:X:POSition:STOP?

PXA command and query:

 :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:RIGHT <freq>

 :CALCulate:MARKer[1]|2|...|12:FUNction:BAND:RIGHT?

In PSA, the command `:CALCulate:MARKer[n]:X:POSition:STOP <param>` is used to control the Reference marker in trace points (buckets) in Band Pair/Delta Pair mode. PXA supports this command for backwards compatibility by first converting the specified value in trace points to the current X Axis Scale Units (frequency or time) of the trace upon which the marker resides. Then, that value is used in the `:CALC:MARKer[n]:FUNction:BAND:RIGHT <param>` command to set the stop position of the marker's Band Function.

Note also that PXA's "trace point" is tied to trace, not to display. When you change the center or span frequency, the trace point table is not updated until you initiate a sweep. This means the trace point and display point are not synchronized, so sending `:CALCulate:MARKer:X:POSition 0` does not move the marker to the left edge of the display. To avoid this problem, perform a sweep, or use `:CALCulate:MARKer:X` instead.

The query returns the marker function RIGHT value in trace points (buckets), by translating the value based on the current X Axis Scale settings.

Marker Commands

Band/Interval Span

Band/Interval Span

PSA command and query, supported in PXA for backwards compatibility:

 :CALCulate:MARKer[1]|2|3|4:X:POSition:SPAN <param>

 :CALCulate:MARKer[1]|2|3|4:X:POSition:SPAN?

PXA command and query:

 :CALCulate:MARKer[1]|2|...|12:FUNCTion:BAND:SPAN <freq>

 :CALCulate:MARKer[1]|2|...|12:FUNCTion:BAND:SPAN?

The PSA command `:CALCulate:MARKer[n]:X:POSition:SPAN <param>` is used to set the span between a delta marker and its reference marker in trace points (buckets) in Span Pair mode. PXA supports this command for backwards compatibility by first converting the specified span in trace points to the current X Axis Scale Units (frequency or time) of the trace upon which the marker resides. Then, that value is sent to the `:CALC:MARKer[n]:FUNCTion:BAND:SPAN <param>` command to set the span of the marker's Band Function.

The query returns the marker function span in trace points (buckets) by translating the value based on the current X Axis Scale settings.

See ["Marker Function Settings" on page 119](#) for more information

Band Span Auto

PXA-only command and query:

 :CALCulate:MARKer[1]|2|...|12:FUNCTION:BAND:SPAN:AUTO ON|OFF

 :CALCulate:MARKer[1]|2|...|12:FUNCTION:BAND:SPAN:AUTO?

In PSA, the Noise Marker width is always equal to 5% of the span. However, in PXA you can change the span of the Marker Noise band using the Band Adjust function. To emulate the PSA behavior, this Band Span Auto/Man function is provided.

When Auto is ON (which it is by default), the Marker Noise band is always held at 5% of Span, even if the Span changes. When you adjust the Marker Noise Band Span, Auto is set to Manual (OFF). Hence the PSA behavior is preserved, but now you can also set the Marker Noise Span, and that setting is preserved when Span is changed.

Continuous Peak Search Setting

PXA command and query:

 :CALCulate:MARKer[1]|2|...|12:CPSearch[:STATE] ON|OFF|1|0

 :CALCulate:MARKer[1]|2|...|12:CPSearch[:STATE]?

PSA command and query, which are **not** aliased in PXA:

 :CALCulate:MARKer[1]|2|3|4:CPEak[:STATE] OFF|ON|0|1

 :CALCulate:MARKer[1]|2|3|4:CPEak[:STATE]?

In ESA and PSA, the Continuous Pk function only considers a peak within a small window relative to the marker's previous position, and thus is designed to track a signal drifting in frequency but with similar amplitude. The PXA Continuous Peak Search function simply performs a Peak Search operation after each sweep, with no regard for the marker's previous position. Because of this difference, the PSA command is **not** accepted by PXA.

Also in ESA and PSA, Continuous Pk is grayed out when span equals zero. The PXA Continuous Peak Search function is available within zero span.

Also in ESA and PSA, turning Continuous Pk on does not automatically execute a peak search. A peak search is not performed until the end of the next sweep. The PXA Continuous Peak Search function performs a peak search when it is turned on, without waiting for the next sweep to complete.

Frequency Counter Gate Time

PSA and PXA commands and queries:

✓ :CALCulate:MARKer[1-4]:FCOunt:GATetime <time>

✓ :CALCulate:MARKer[1-4]:FCOunt:GATetime?

✓ :CALCulate:MARKer[1-4]:FCOunt:GATetime:AUTO OFF|ON|0|1

✓ :CALCulate:MARKer[1-4]:FCOunt:GATetime:AUTO?

For applicability of these commands to the selected measurement, see [“Marker Commands & PowerSuite Measurements”](#) on page 94.

Marker Commands

Frequency Counter State

Frequency Counter State

PSA and PXA commands and queries:

 :CALCulate:MARKer[1]|2|...|12:FCOunt[:STATe] OFF|ON|0|1

 :CALCulate:MARKer[1]|2|...|12:FCOunt[:STATe]?

 :CALCulate:MARKer[1]|2|...|12:FCOunt:X?

In ESA and PSA, the reference marker for Delta markers is always counted.

In PXA, the marker is counted for Normal and Delta markers; but if the reference marker is a Fixed marker, the count stored in the Fixed marker is used. This enhanced capability may require changes in code written for PSA.

For applicability of these commands to the selected measurement, see [“Marker Commands & PowerSuite Measurements” on page 94](#).

Marker Delta to Span

PSA and PXA command:

 :CALCulate:MARKer[1]|2|...|12[:SET]:DELta:SPAN

In ESA and PSA, Mkr Δ ->Span adjusts the span and changes the Delta marker to a Normal marker, placing it at the center of the screen.

In PXA, the markers remain in Delta mode, and the Delta and Reference marker end up on opposite edges of the display.

Marker Commands

Marker Maximum and Minimum (CCDF Measurement)

Marker Maximum and Minimum (CCDF Measurement)

 :CALCulate:PStAtistic:MARKer[1]|2|3|4:MAXimum

 :CALCulate:PStAtistic:MARKer[1]|2|3|4:MINimum

 [:SENSE]:PStAtistic:MARKer[1]|2|3|4:MAXimum

 [:SENSE]:PStAtistic:MARKer[1]|2|3|4:MINimum

These PSA commands are **not** supported in PXA, and are not aliased to any PXA commands.

Marker Table

PSA and PXA command and query (STATE keyword optional only in PXA):

 :CALCulate:MARKer:TABLE[:STATE] OFF|ON|0|1

 :CALCulate:MARKer:TABLE[:STATE]?

In PSA, this functionality is available in all PowerSuite measurements via SCPI, even though the CCDF, Spectrum Emission Mask and Spurious Emissions measurements do not have related menus. When Marker Table is switched on, the measurement is switched to SA before the Marker Table is displayed.

If you change the measurement from SA to any other measurement while Marker Table is on, the Marker Table is automatically turned off.

In PXA, this command is available only in the Swept SA measurement.

Marker Commands
Marker to Reference Level

Marker to Reference Level

PSA and PXA command:

 :CALCulate:MARKer[1] | 2 | . . . | 12[:SET]:RLEVel

In PXA, the Mkr-> RefLvl behavior for a delta marker differs slightly from that in PSA. PSA assigns the delta marker's amplitude to the reference level, ignoring the reference marker altogether. PXA allows you to select either the reference or the delta marker individually, then the selected marker's amplitude is applied to the reference level.

Marker Trace Auto

PSA and PXA commands and queries:

 :CALCulate:MARKer[1]|2|...|12:TRACe:AUTO OFF|ON|0|1

 :CALCulate:MARKer[1]|2|...|12:TRACe:AUTO?

The Marker Trace Auto function in PSA has been replaced by Marker Trace Auto Init, but the same SCPI command is used for the new function.

The PXA auto functionality causes markers to automatically go to the appropriate trace when the marker is first switched on. Thus, any PSA code that relies on markers changing traces when a trace is moved into out of View may require modification.

Marker Commands

Marker Type (CCDF Measurement)

Marker Type (CCDF Measurement)

PSA command:

 :CALCulate:PStatistic:MARKer[1]|2|3|4:MODE NORMAl|DELTA|OFF

PXA command:

 :CALCulate:PStatistic:MARKer[1]|2|...|12:MODE POSition|DELTA|OFF

Note the difference in the enum parameter selection between PSA and PXA, which could cause a code compatibility problem.

PSA command and query:

 :CALCulate:PStatistic:MARKer[1]|2|3|4[:STATe] OFF|ON|0|1

 :CALCulate:PStatistic:MARKer[1]|2|3|4[:STATe]?

In PXA, these commands are supported for backwards compatibility, but the STATe keyword is **not** optional.

Marker Type (Spectrum Emission Mask Measurement)

PSA command and query, supported in PXA for backwards compatibility:

✓ :CALCulate:SEMask:MARKer[1]|2|3|4[:STATE] OFF|ON|0|1

✓ :CALCulate:SEMask:MARKer[1]|2|3|4[:STATE]?

Equivalent PXA command and query:

✓ :CALCulate:SEMask:MARKer[1]|2|...|12:MODE POSITION|OFF

✓ :CALCulate:SEMask:MARKer[1]|2|...|12:MODE?

Marker Commands

Marker Type (Spurious Emissions Measurement)

Marker Type (Spurious Emissions Measurement)

PSA command and query, supported in PXA for backwards compatibility:

✓ :CALCulate:SPURious:MARKer[1]|2|3|4[:STATE] OFF|ON|0|1

✓ :CALCulate:SPURious:MARKer[1]|2|3|4[:STATE]?

Equivalent PXA command and query:

✓ :CALCulate:SPURious:MARKer[1]|2|...|12:MODE POSITION|DELTA|OFF

✓ :CALCulate:SPURious:MARKer[1]|2|...|12:MODE?

Next Peak Commands

PSA and PXA commands, but see notes below for applicability:

 :CALCulate:MARKer[1]|2|...|12:MAXimum:NEXT

 :CALCulate:MARKer[1]|2|...|12:MAXimum:RIGHT

 :CALCulate:MARKer[1]|2|...|12:MAXimum:LEFT

For applicability of these commands to the selected measurement, see [“Marker Commands & PowerSuite Measurements” on page 94](#).

In PXA, only the Swept SA measurement supports the Next Peak, Next Right, Next Left search functionality. To emulate these search functions in other measurements, obtain a peak list sorted by amplitude using :CALCulate:DATA[1]|2|3|4|5|6:PEAKs? <threshold>, <excursion>, AMPLitude, then find the next biggest peak from the current marker position (then move the marker if necessary).

When you perform a Next Left or Next Right search in PSA, the Peak Excursion setting is applied only to the left or right side of the current marker position. Conversely, PXA always applies the setting to the entire trace regardless of the current marker position. If a marker is positioned near peak, and amplitude delta is less than peak excursion, PSA ignores the near peak, and looks for the next biggest peak, whereas PXA finds the near peak. To emulate PSA behavior in PXA, use the following algorithm:

1. Acquire the current marker position x_1 ,
2. Perform a Next Peak Left or Right operation,
3. Acquire the current marker position x_2 ,
4. Acquire the current marker amplitude y_2 ,
5. Acquire the minimum value y_{min} between x_1 and x_2 using CALCulate:DATA:COMPRESS? MINimum, x_1 , x_2-x_1 ,
6. If delta between y_{min} and y_2 is less than the peak excursion value, perform a Next Peak Left or Right operation again.

Marker Commands

Peak Search, Min Search, Peak to Peak, Marker Trace

Peak Search, Min Search, Peak to Peak, Marker Trace

PSA and PXA commands:

 :CALCulate:MARKer[1]|2|...|12:MAXimum

 :CALCulate:MARKer[1]|2|...|12:MINimum

 :CALCulate:MARKer[1]|2|...|12:PTPeak

 :CALCulate:MARKer[1]|2|...|12:TRACe 1|...|6

For applicability of these commands to the selected measurement, see [“Marker Commands & PowerSuite Measurements”](#) on page 94.

Signal Track

PSA command and query, supported in PXA **without** marker subopcode ([1] | 2 | 3 | 4):

 :CALCulate:MARKer[1] | 2 | 3 | 4:TRCKing[:STATE] OFF|ON|0|1

 :CALCulate:MARKer[1] | 2 | 3 | 4:TRCKing[:STATE]?

PXA command and query:

 :CALCulate:MARKer:TRCKing[:STATE] OFF|ON|0|1

 :CALCulate:MARKer:TRCKing[:STATE]?

In PXA, Signal Track is bound only to **Marker 1**, and cannot be enabled for any other marker, whereas PSA permits a subopcode specifying which marker to use.

PXA supports this functionality only in the Swept SA measurement.

In PXA, Signal Track is in the Span menu, whereas in PSA it is located in the Frequency menu.

Marker Commands

X Axis Scale (Readout)

X Axis Scale (Readout)

PSA and PXA commands and queries:

 :CALCulate:MARKer[1]|2|...|12:X:READout FREQuency|TIME|ITIME|PERiod

 :CALCulate:MARKer[1]|2|...|12:X:READout?

 :CALCulate:MARKer[1]|2|...|12:X:READout:AUTO ON|OFF|1|0

 :CALCulate:MARKer[1]|2|...|12:X:READout:AUTO?

In PXA, the X Axis Scale of a marker (**Readout** in PSA) has only one value, instead of one value for frequency domain and another value for time domain. If in Auto, the value changes whenever the domain of the trace it is on changes. This means that the default behaviors are identical, but if code intended for PSA changes the readout manually in Swept, assuming that the value will remain the same in Zero Span, there may be a backwards compatibility problem.

For example, in PSA, if you set Readout to Period in a Swept span, and the instrument is set to Zero Span, the readout changes to Time, the default for Zero Span. In PXA, it stays in Period even in Zero Span until the you change it or set it back to Auto.

Additionally, in PXA all choices for X Axis Scale are always permitted, whereas in PSA the choices of X Axis Scale are restricted based on the current domain of the instrument.

Marker Function Settings

 :CALCulate:MARKer[1]|2|...|12:FUNCTION NOISE|BPOWER|BDENSITY|OFF

 :CALCulate:MARKer[1]|2|...|12:FUNCTION?

See [“Band Power Functionality & Marker Modes” on page 93](#).

For applicability of these commands to the selected measurement, see [“Marker Commands & PowerSuite Measurements” on page 94](#).

Direct Marker Positioning

The following commands are used in ESA and PSA to directly set the marker to a specific trace point (“bucket”) position when they are being used in Span Pair and Delta Pair/Band Pair modes:

 :CALCulate:MARKer[1]|2|3|4:X:POSITION:CENTer <param>

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

 :CALCulate:MARKer[1]|2|3|4:X:POSITION:SPAN <param>

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

 :CALCulate:MARKer[1]|2|3|4:X:POSITION:START <param>

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

 :CALCulate:MARKer[1]|2|3|4:X:POSITION:STOP <param>

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

In PSA, these commands are aliased very similarly to the non-position commands (see [“Span Pair Commands” on page 98](#) and [“Delta Pair \(Band Pair\) Commands” on page 99](#)), except that in these cases a translation to/from trace points (buckets) is also performed:

Old command	Aliased to
:CALCulate:MARKer[1] 2 3 4:X:POSITION:CENTer	:CALCulate:MARKer[1] 2 3 4:X:POSITION
:CALCulate:MARKer[1] 2 3 4:X:POSITION:SPAN	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:SPAN
:CALCulate:MARKer[1] 2 3 4:X:POSITION:START	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:LEFT
:CALCulate:MARKer[1] 2 3 4:X:POSITION:STOP	:CALCulate:MARKer[1] 2 3 4:FUNCTION:BAND:RIGHT

Note that all of the alias commands described here cause the specified marker to become selected.

In each case except :CALCulate:MARKer[1]|2|3|4:X:POSITION:CENTer, the instrument first converts the specified value in trace points to the current X Axis Scale Units (frequency or time) of the trace upon which the marker resides. Then, that value is used in the alias command to set the desired value.

The query returns the marker function span in trace points (buckets) by translating the value based on the current X Axis Scale settings.

Marker Commands

Marker Function Settings

The value in Trace Points is translated into the current X Axis Scale units for the purpose of setting the value of the marker. However, the marker's span value, LEFT value, or RIGHT value in X Axis Scale Units, **not** trace points, is preserved if a change is made to the X Axis scale settings. For example, if you use this command to set a marker function span of 500 buckets, which happens at that time to correspond to 13 GHz, and then you change the analyzer's Start Frequency so that 500 buckets is no longer 13 GHz, the span will stay at 13 GHz, **not** at 500 buckets! This is important to realize as it differs from the legacy behavior.

The UP/DOWN parameters increment/decrement by one bucket. For this the instrument performs a conversion to buckets and back.

Band Adjust

If any of the band adjust SCPI commands are sent while the marker function is off, they are accepted and the value is stored. If sent while the marker is on, they are accepted but the value is ignored and discarded.

Detector Dwell Times

PXA command and query:

 :CALCulate:MAMarker:DETEctor[1]|2|3:DWELL <dwell time>

 :CALCulate:MAMarker:DETEctor[1]|2|3:DWELL?

PSA command and query, supported in PXA for backwards compatibility:

 [:SENSe]:EMI:MEASure:DETEctor:DWELL <dwell time>

This command is included for compatibility with PSA option 239. It sets the dwell time for all of the detectors to the specified value.

Marker Commands
Measure at Marker Query

Measure at Marker Query

PXA query:

 :CALCulate:MARKer[1]|2|...|12:FUNCTION:MAMarker?

PSA query, supported in PXA for backwards compatibility:

 :MEASure:EMI:MARKer[1]|2|...|12?

This query is supported for compatibility with PSA option 239. It causes the instrument to perform a Measure at Marker function at the specified marker's current frequency, then return the results.

Preselector Automatic Centering

PXA command and query:

 :CALCulate:MAMarker:PCENTER ON|OFF|1|0

 :CALCulate:MAMarker:PCENTER?

PSA command and query, supported in PXA for backwards compatibility:

 [:SENSe]:EMI:MEASure:PCENTER[:STATe] OFF|ON|0|1

 [:SENSe]:EMI:MEASure:PCENTER[:STATe]?

This command is included for compatibility with PSA option 239.

Marker Commands
Preselector Automatic Centering

11 Span / X Scale Commands

Scale/Div (CCDF Measurement)

PSA command and query, supported in PXA for backwards compatibility:

✓ `:DISPlay:PSStatistic:XScale <rel_ampl>`

✓ `:DISPlay:PSStatistic:XScale?`

PXA command and query:

✓ `:DISPlay:PSStatistic:VIEW[1]:WINDow2:TRACe:X[:SCALE]:PDIVision <rel_ampl>`

✓ `:DISPlay:PSStatistic:VIEW[1]:WINDow2:TRACe:X[:SCALE]:PDIVision?`

Span (Channel Power Measurement)

PXA and PSA command:

⚠ `[:SENSe] :CHPower:FREQuency:SPAN <freq>`

In PSA, this parameter is completely coupled with `[:SENSe] :FREQuency:SPAN`.

In PXA, this parameter is local to the Channel Power measurement. The command `[:SENSe] :FREQuency:SPAN` applies only to the Swept SA measurement.

Span (Occupied Band width Measurement)

⚠ `[:SENSe] :OBWidth:FREQuency:SPAN <freq>`

⚠ `[:SENSe] :OBWidth:FREQuency:SPAN?`

In PSA, this parameter is completely coupled with `[:SENSe] :FREQuency:SPAN`.

In PXA, this parameter is local to the OBWidth measurement. The command `[:SENSe] :FREQuency:SPAN` applies only to the Swept SA measurement.

Span / X Scale Commands
Span (Occupied Bandwidth Measurement)

12 Sweep Commands

External Gate Level

PSA command and query also supported by PXA for backwards compatibility:

✓ `[:SENSe] :SWEep:EGATe:EXTeRnal[1] | 2:LEVe1 <voltage>`

✓ `[:SENSe] :SWEep:EGATe:EXTeRnal[1] | 2:LEVe1?`

Sets the gate input transition point level for the external Trigger inputs on the front and rear panel.

In PXA, the command is aliased to `:TRIGger[:SEQuence] :EXTeRnal[1] | 2:LEVe1`.

Sweep Commands

Gate Source

Gate Source

PSA command:

 [:SENSe]:SWEep:EGATe:SOURce EXTernal[1]|EXTernal2|RFBurst

PXA command:

 [:SENSe]:SWEep:EGATe:SOURce EXTernal1|EXTernal2|LINE|FRAME|RFBurst

PSA and PXA query:

 [:SENSe]:SWEep:EGATe:SOURce?

In PSA, the Gate Source may be taken from one of two specified input ports.

In PXA, any Trigger Source may be a Gate Source, hence the additional options for this command.

FFT Width

PXA command and query:

 [:SENSe]:SWEep:FFT:WIDTh <real>

 [:SENSe]:SWEep:FFT:WIDTh?

PSA command and query also supported by PXA for backwards compatibility*:

 [:SENSe]:SWEep:FFT:SPAN:RATio <integer>

 [:SENSe]:SWEep:FFT:SPAN:RATio?

(* In PXA, for Swept SA measurement, this command is aliased to [:SENSe]:SWEep:FFT:WIDTh. Since other measurements do not support user-controllable FFT sweep, this command is not supported in those measurements.)

This is the legacy “FFTs per Span” command, because in the PSA, this is what you set rather than the FFT Width. The behavior of the analyzer when it receives this command is to compute the “intended segment width” by dividing the Span by the FFTs/Span parameter, then converting this intended width to an actual width by using the largest available FFT Width that is still less than the intended segment width. The “Span” used in this computation is whatever the Span is currently set to, whether a sweep has been taken at that Span or not.

In PSA, since the Retain Parameter setting has no effect on this parameter; the value is reset always on measurement switch.

Sweep Commands

Sweep Points

Sweep Points

PSA and PXA command and query:



`[:SENSe] :SWEep:POINTs <integer>`



`[:SENSe] :SWEep:POINTs?`

In ESA and PSA, Sweep Points is adjustable with the knob and step keys. This causes the sweep time to increase whenever Points is adjusted (either up or down). In PXA, the value of Sweep Points must be entered manually, which avoids this anomaly.

In PSA the preset value for Sweep Points is 601. In PXA, it is 1001. To set the value to 601 for PXA, send `:SWEep:POINTs 601` after every preset.

In PSA, the default RBW/VBW ratio is 1, while in PXA it is 0.1. This sets a narrower Video Bandwidth for PXA, which could result in a slower measurement.

Note that, even if you set the same RBW/VBW ratio, the timing for Gate measurements may differ between PSA and PXA.

The Number of Trace Points setting on PSA is measurement global (except for Spurious Emissions and CCDF measurements), whereas PXA stores trace points by measurement. This may cause a problem if code written for PSA sets the number of points in, for example, the SA measurement then expects the number of points to remain the same in another measurement, such as Channel Power.

Sweep Time

PSA and PXA commands and queries:

 [:SENSe] :SWEep:TIME <time>

 [:SENSe] :SWEep:TIME?

 [:SENSe] :SWEep:TIME:AUTO OFF|ON|0|1

 [:SENSe] :SWEep:TIME:AUTO?

In PSA, this functionality is affected by the Retain Parameter setting. When Retain Parameter is on, the value is retained between measurement switches. When Retain Parameter is off, the value is reset to its default on a measurement switch.

In the Harmonics measurement the value is automatically determined, and cannot be set. In the Spurious Emissions and Spectrum Emission Mask measurements, the value is overridden by the value defined in the Range Table, and cannot be defined via this command.

In PXA, this setting is stored per measurement.

In PXA, when in zero span, changing the sweep time moves marker display position, whereas in PSA the position does not change.

In PXA, sending `SWEep:TIME:AUTO ON` when FFT is enabled generates the error message "Setting Conflict". PSA does not generate this error.

Sweep Commands

Sweep Time Rules

Sweep Time Rules

PXA command and query:



[:SENSe] :SWEep:TIME:AUTO:RULEs NORMAl | ACCuracy | SRESponse



[:SENSe] :SWEep:TIME:AUTO:RULEs?

PSA command and query also supported by PXA for backwards compatibility, and aliased as described:



[:SENSe] :SWEep:TIME:AUTO:MODE SRESponse | SANalyzer

In PXA, this command with the SRESponse option is aliased to [:SENSe] :SWEep:TIME:AUTO:RULEs SRESponse. With the SANalyzer option, it is aliased to :SWEep:TIME:AUTO:RULEs NORMAl.



[:SENSe] :SWEep:TIME:AUTO:MODE?

In PXA, this query is aliased to [:SENSe] :SWEep:TIME:RULEs?, so the response matches PSA for SRESponse, but not for SANalyzer.

In PSA, the Retain Parameter setting has no effect on this parameter, so the value is reset always on a measurement switch. However, the CCDF, Harmonics, and Burst Power measurements use a fixed value for the parameter.

In PXA, this setting is stored per measurement, except that the Spectrum Emission Mask measurement does not support it.

Sweep Type

PXA commands and queries:

 [:SENSe] :SWEep:TYPE FFT | SWEep

 [:SENSe] :SWEep:TYPE:AUTO OFF | ON | 0 | 1

 [:SENSe] :SWEep:TYPE:AUTO?

PSA command, accepted by PXA:

 [:SENSe] :SWEep:TYPE AUTO | FFT | SWEep

Sending the PSA command [:SENSe] :SWEep:TYPE AUTO to PXA sets sweep type Auto to on, but the query returns either FFT or SWE, depending on the auto setting.

PXA and PSA query:

 [:SENSe] :SWEep:TYPE?

In PSA, the Retain Parameter setting has no effect on this parameter; the value is reset always on a measurement switch.

In PXA, only the Swept SA measurement supports this functionality.

For additional backwards compatibility, PXA accepts the command [:SENSe] :SWEep:TYPE SWP, and selects sweep type SWEep.

Sweep Commands
Sweep Type

13 Trace/Detector Commands

Detector

PXA command and query:

✓ [:SENSe]:DETECTOR:TRACe[1]|2|...6
AVERAge|NEGAtive|NORMAl|POSitive|SAMPle|QPEak|EAVERage|RAVERage

✓ [:SENSe]:DETECTOR:TRACe[1]|2|...6?

PSA command and query, supported in PXA for backwards compatibility (affects all traces):

⚠ [:SENSe]:DETECTOR[:FUNCTION]
NORMAl|AVERAge|POSitive|SAMPle|NEGAtive|QPEak|EAVERage|EPOSitive|MPOSitive|RMS

⚠ [:SENSe]:DETECTOR[:FUNCTION]?

The RMS, EPOSitive and MPOSitive parameter options of the backwards compatibility command are mapped as shown in “[Command Parameter Mapping](#)” on page 135 below. The backwards compatibility query returns a value that corresponds to the detector type as shown in “[Command Parameter Mapping](#)” on page 135 below, and indicates the setting for Trace 1.

When averaging is switched on, PSA selects AVERAge detector mode, whereas PXA selects SAMPle detector mode. Therefore, in PXA, AVERAge mode must be selected explicitly after averaging is switched on.

In PSA, this functionality is affected by the Retain Parameter setting. When Retain Parameter is on, the value is retained between measurement switches. When Retain Parameter is off, the value is reset to its default on a measurement switch.

In PSA’s Harmonic Distortion, Spurious Emissions and Spectrum Emission Mask measurements, the value is overridden by the value defined in Range Table. Additionally, in the Harmonic Distortion and Spurious Emissions measurements, the value cannot be set via this command.

In PXA, this functionality is supported as measurement local parameter.

Command Parameter Mapping

PXA maps parameters of the backwards compatibility command ([:SENSe]:DETECTOR[:FUNCTION]) as follows:

Parameter	Settings	Query Returns
RMS	Sets detector type to AVERAge and Average Type to RMS.	AVER
EPOSitive	Sets detector type to Peak and EMI Standard to CISPR.	POS

Trace/Detector Commands
Detector

Parameter	Settings	Query Returns
MPOsitive	Sets detector type to Peak and EMI Standard to MIL Impulse.	POS

Detector Auto Setting

PXA command and query:

 `[:SENSe]:DETECTOR:TRACe[1]|2|...6:AUTO ON|OFF|1|0`

 `[:SENSe]:DETECTOR:TRACe[1]|2|...6:AUTO?`

PSA command and query, supported in PXA for backwards compatibility (affects all traces):

 `[:SENSe]:DETECTOR:AUTO ON|OFF|1|0`

 `[:SENSe]:DETECTOR:AUTO?`

In PSA, this functionality is affected by the Retain Parameter setting. When Retain Parameter is on, the value is retained between measurement switches. When Retain Parameter is off, the value is reset to its default on a measurement switch.

In PXA, this functionality is supported as measurement local parameter.

Normalize Trace Data Settings (State, Reference Level, Reference Position)

PSA and PXA commands and queries:

 `:CALCulate:NTData[:STATE] OFF|ON|0|1`

 `:CALCulate:NTData[:STATE]?`

 `:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:NRLevel <rel_ampl>`

 `:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:NRLevel?`

 `:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:NRPosition <integer>`

 `:DISPlay:WINDow[1]:TRACe:Y[:SCALE]:NRPosition?`

In PSA, the applicability of these commands depends on the selected measurement. In the CCDF and Spectrum Emission Mask measurements, the Normalize functionality is not supported and the commands have no effect.

In the Spurious Emissions measurement, the functionality may be switched on via SCPI, even though there is no related menu.

In all other PSA measurements, the Normalize settings act as measurement global parameters.

In PXA, only the Swept SA measurement supports this functionality.

Trace/Detector Commands

Trace Mode (Type)

Trace Mode (Type)

PXA command and query:

✓ :TRACe[1] | 2 | ... 6 :TYPE WRITe | AVERAge | MAXHold | MINHold

✓ :TRACe[1] | 2 | ... 6 :TYPE?

PSA command and query, supported in PXA for backwards compatibility (for Swept SA measurement only):

⚠ :TRACe[1] | 2 | ... | 6 :MODE WRITe | MAXHold | MINHold | VIEW | BLANK

⚠ :TRACe[1] | 2 | ... | 6 :MODE?

In PSA, :TRACe[n]:MODE is used to set the type or “writing mode” of the trace. The writing modes include **VIEW** and **BLANK**. The PXA :TRACe[n]:TYPE command should be used for new code, but :TRACe[n]:MODE is retained to provide backwards compatibility.

When sent to PXA, the :TRACe[n]:MODE? query returns the same value as :TRACe[...]:TYPE?, so it never returns the values **VIEW** or **BLANK**.

The following associated PSA command and query are supported in PXA for backwards compatibility (for Swept SA measurement only):

⚠ [:SENSe] :AVERAge [:STATe] ON | OFF | 1 | 0

⚠ [:SENSe] :AVERAge [:STATe] ?

In PSA, Averaging (also known as trace averaging) is global to all traces, that is, it is either on or off for all active traces. The command [:SENSe] :AVERAge [:STATe] is used to switch averaging on or off.

In PXA, Averaging is turned on or off on a per-trace basis, so it can be on for one trace, and off for another.

PXA retains the PSA global Average State setting solely as a legacy variable, switched on and off and queried by the legacy command [:SENSe] :AVERAge [:STATe].

When received by PXA, the :TRACe[n]:MODE options affect settings as follows:

PSA Command	:TRACe[n]:TYPE Equivalent Option	PXA :TRACe[n]:UPDate Setting	PXA :TRACe[n]:DISPlay Setting
:TRACe:MODE WRITe	WRITe, unless Average is true, in which case AVERAge.	ON 1	ON 1
:TRACe:MODE MAXHold	MAXHold	ON 1	ON 1
:TRACe:MODE MINHold	MINHold	ON 1	ON 1

For :TRACe:MODE VIEW and :TRACe:MODE BLANK, see “[View & Blank Options](#)” on page 139 below.

In PSA, the functionality described above is supported by all measurements except CCDF, Harmonic Distortion, Spurious Emissions, and Spectrum Emission Mask.

MAXHold & MINHold Issues

In PSA, **MAXHold** and **MINHold** are conditions of the trace and keep holding continuously. Conversely, in PXA **MAXHold** and **MINHold** are averaging types, which take account of the averaging count.

In the following code sample, `:INITiate:IMMediate` is followed by an `*OPC?` query while Max Hold is on, In PSA `*OPC?` returns 1 after the **first** sweep, whereas PXA waits for 100 sweeps (by default) and thus takes much longer to complete than PSA.

```
*RST
:INITiate:CONTInuous OFF
:TRACe:MODE MAXHold
:INITiate:IMMediate; *OPC?
```

To emulate the PSA behavior in PXA, set Average Count = 1 (by sending `[:SENSe]:AVERAge:COUNT 1`).

View & Blank Options

In PSA, `VIEW` and `BLANK` are trace modes, set via the `:TRACe[n]:MODE` command.

Conversely, in PXA, `VIEW` and `BLANK` map to trace update and display settings, which may be set or queried via new `:TRACe[n]:UPDate` and `:TRACe[n]:DISPlay` commands, as listed below:

 `:TRACe[1]|2|...|6:UPDate[:STATe] ON|OFF|0|1`

 `:TRACe[1]|2|...|6:UPDate[:STATe]?`

 `:TRACe[1]|2|...|6:DISPlay[:STATe] ON|OFF|0|1`

 `:TRACe[1]|2|...|6:DISPlay[:STATe]?`

The table below shows how the PSA `VIEW` and `BLANK` trace mode options are mapped in PXA to the settings controlled by the new trace update and display commands:

PSA Command	PXA :TRACe[n]:UPDate Setting	PXA :TRACe[n]:DISPlay Setting
<code>:TRACe[n]:MODE VIEW</code>	OFF 0	ON 1
<code>:TRACe[n]:MODE BLANK</code>	OFF 0	OFF 0

Trace Data



:TRACe[1] | 2 | 3[:DATA] TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6, <data>



:TRACe[1] | 2 | 3[:DATA]? TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6



:TRACe[1] | 2 | 3[:DATA]? TRACE[n]

Note that, in PXA, the options RAWTRACE, LLINE1, and LLINE2 for the trace data query are no longer supported.

Measurement Setting

In PSA, trace data may in general always be retrieved using the TRACe[:DATA]? query, regardless of whether the instrument is set to the SA measurement or any other PowerSuite measurement. However, the query returns unavailable data when the CCDF measurement is selected.

In PXA, the :TRACe[:DATA]? query always returns trace data from the **Swept SA** measurement. To query trace data from any other PowerSuite measurement, use :READ:<meas>[n]? instead, where the value of n depends on the measurement as listed below.

Measurement	<meas>	n
Adjacent Channel Power	ACPower	4, 5, 6
Burst Power	BPOwer, TXPower	2
Channel Power	CHPower	2
Occupied Bandwidth	OBWidth	2
Spectrum Emission Mask	SEMask	2

Data Format

When ASCII data format is selected, the length of trace data points returned differs as described in the table below.

Instrument Type	Data Length	Digits following Decimal Point	Example
PSA	12 bytes	5	-4.02083E+01
PXA	15 or 16 bytes. See below for details.	9	-2.256545976E+01

PSA inserts a + sign for positive values, so the data length is fixed at 12 bytes per point, regardless of the sign of the data.

PXA does not insert a + sign for positive values, so the data length varies (16 bytes for negative values or 15 bytes for positive values). Thus, if code written for PSA anticipates fixed length data, this may create a problem.

Trace Copy & Exchange

 :TRACe[1] | 2 | 3 : COPY TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6,
TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6

 :TRACe[1] | 2 | 3 : COPY?

 :TRACe[1] | 2 | 3 : EXCHange TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6,
TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6

 :TRACe[1] | 2 | 3 : EXCHange?

In PSA, this functionality is supported in all measurements except CCDF, Harmonic Distortion, Spectrum Emission Mask and Spurious Emissions.

In PXA, only the Swept SA measurement supports this functionality.

Trace/Detector Commands
Trace Math Add & Subtract

Trace Math Add & Subtract

PSA commands:

 :TRACe[1] | 2 | 3 :MATH:ADD <dest_trace> , <src_trace> , <src_trace>

 :TRACe[1] | 2 | 3 :MATH:SUBTRACT <dest_trace> , <src_trace> , <src_trace>

 :TRACe[1] | 2 | 3 :MATH:SUBTRACT:DLINe <trace_name>

These PSA commands are no longer available in PXA.

Mean Trace Data Query

PSA query, supported in PXA for backwards compatibility:



`:TRACe[1] | 2 | 3 :MATH:MEAN? TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6`

Use of this query is not recommended for new designs. Use the `:CALCulate:DATA:COMPRESS?` query instead.

In PSA, this query is supported in every measurement; although it returns unavailable data for the CCDF measurement.

In PXA, only the Swept SA measurement supports this functionality.

Peak Data & Peak Points Queries

PSA queries, supported in PXA for backwards compatibility:



:TRACe[1] | 2 | 3 :MATH:PEAK[:DATA]?



:TRACe[1] | 2 | 3 :MATH:PEAK:POINTs?

Use of these queries is not recommended for new designs. Use the :CALCuLate:DATA[1] | 2 | 3 | 4 | 5 | 6 :PEAKs? query instead.

Trace Data Smooth & Points for Smoothing

PSA commands, supported in PXA for backwards compatibility:

 :TRACe[1] | 2 | 3 :MATH:SMOoth TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 | TRACE6

 :TRACe[1] | 2 | 3 :MATH:SMOoth:POINTs <integer>

 :TRACe[1] | 2 | 3 :MATH:SMOoth:POINTs?

Use of these commands is not recommended for new designs. Use the :CALCulate:DATA:COMPRESS? command instead.

Trace/Detector Commands
Trace Data Smooth & Points for Smoothing

14 Trigger Commands

Trigger Source

PSA command and query:

 :TRIGger[:SEQuence]:SOURce IMMEDIATE|VIDeo|LINE|EXTErnal[1]|EXTErnal2|RFBurst

 :TRIGger[:SEQuence]:SOURce?

PXA command and query:

 :TRIGger:<meas>[:SEQuence]:SOURce
EXTErnal1|EXTErnal2|IMMEDIATE|LINE|FRAME|RFBurst|VIDeo|IF|ALARm|LAN|IQMag|
IDEMod|QDEMod|IINPut|QINPut|AIQMag|TV

 :TRIGger:<meas>[:SEQuence]:SOURce?

In PSA, the trigger source setting is a measurement global parameter. The same setting applies to SA and all PowerSuite measurements.

In PXA, the trigger source setting is local for each measurement. Include the <meas> keyword in the command to specify the measurement for which you wish to set the trigger source. If the <meas> keyword is omitted (as for the PSA command), the setting applies only to the Swept SA measurement.

PXA supports the command :TRIGger[:SEQuence]:SOURce EXTErnal for backwards compatibility. In this case, the parameter EXTErnal is mapped to EXTErnal1.

The PSA command [:SENSe]:<meas>:TRIGger:SOURce is supported by PXA for backwards compatibility, but **not** for the Swept SA, Monitor Spectrum, Log Plot or Spot Frequency measurements. If you specify the IF option with this command, PXA selects VIDeo triggering.

PXA also supports the backwards compatibility command [:SENSe]:ACPr:TRIGger:SOURce, which is mapped to [:SENSe]:ACPr:TRIGger:SOURce for PSA's W-CDMA, cdma2000 and 1xEVDO modes.

Trigger Commands

Trigger Delay

Trigger Delay

PXA commands and queries:

✔ :TRIGger[:SEquence]:DELay <time>

✔ :TRIGger[:SEquence]:DELay?

✔ :TRIGger[:SEquence]:DELay:STATe OFF|ON|0|1

✔ :TRIGger[:SEquence]:DELay:STATe?

PSA command supported in PXA for backwards compatibility:

⚠ :TRIGger[:SEquence]:IF:DELay <time>

Note that in PSA's CCDF measurement, trigger delay cannot be switched on (STATe command). Conversely, in PXA trigger delay is available in all measurements.

In PSA, the Trigger Delay is global to all triggers, whereas in PXA, the delay can be set individually for each Trigger Source. When sent to PXA, the PSA command :TRIGger[:SEquence]:DELay affects the delay for the VIDEO, LINE, EXTERNAL1, and EXTERNAL2 triggers.

The query returns the trigger delay setting of the currently selected trigger source.

External 1, External 2 & Line Trigger Delay

PXA commands and queries:

- ✓ :TRIGger[:SEquence]:EXTErnal1|2:DELAy <time>
- ✓ :TRIGger[:SEquence]:EXTErnal1|2:DELAy?
- ✓ :TRIGger[:SEquence]:EXTErnal1|2:DELAy:STATe OFF|ON|0|1
- ✓ :TRIGger[:SEquence]:EXTErnal1|2:DELAy:STATe?
- ✓ :TRIGger[:SEquence]:EXTErnal:DELAy <time>
- ✓ :TRIGger[:SEquence]:LINE:DELAy <time>
- ✓ :TRIGger[:SEquence]:LINE:DELAy?
- ✓ :TRIGger[:SEquence]:LINE:DELAy:STATe OFF|ON|0|1
- ✓ :TRIGger[:SEquence]:LINE:DELAy:STATe?

For backward compatibility, PXA supports the PSA option `EXTErnal` and maps it to `EXTErnal1`.

Note that, when sent to PXA, the PSA `:TRIGger[:SEquence]:DELAy` command affects the delay for the `VIDeo`, `LINE`, `EXTErnal1`, and `EXTErnal2` triggers. The same is true for the ESA `:TRIGger[:SEquence]:OFFSet` command, when received by PXA (see also ["Video Trigger Delay" on page 153](#)).

Trigger Commands

External 1 Trigger Level

External 1 Trigger Level

PXA command and query:

✔ :TRIGger[:SEquence]:EXTerna1:LEVel <level>

✔ :TRIGger[:SEquence]:EXTerna1:LEVel?

For backwards compatibility, PXA supports the option `EXTerna1` and maps it to `EXTerna11`.

External 1, External 2 & Line Trigger Slope

PXA commands and queries:

✓ :TRIGger[:SEquence]:EXTErnal1|2:SLOPe POSitive|NEGative

✓ :TRIGger[:SEquence]:EXTErnal1|2:SLOPe?

✓ :TRIGger[:SEquence]:LINE:SLOPe POSitive|NEGative

✓ :TRIGger[:SEquence]:LINE:SLOPe?

For backwards compatibility, PXA supports the option **EXTErnal** and maps it to **EXTErnal1**.

In PSA, the Trigger Slope is global to all triggers, whereas in PXA, the slope can be set individually for each Trigger Source. When sent to PXA, the PSA :TRIGger[:SEquence]:SLOPe command affects the slope for the VIDEO, LINE, EXTErnal1, EXTErnal2, and RFBurst triggers.

The query returns the trigger slope setting of the currently selected trigger source.

Trigger Commands

RF Burst Trigger Level (Relative)

RF Burst Trigger Level (Relative)

PXA commands and queries:

 :TRIGger[:SEquence]:RFBurst:LEVel:RELative <rel_ampl>

 :TRIGger[:SEquence]:RFBurst:LEVel:RELative?

PSA command supported in PXA for backwards compatibility:

 :TRIGger[:SEquence]:RFBurst:LEVel <rel_ampl>

PSA has **only** relative burst triggering, so in PXA this command is aliased to :TRIGger[:SEquence]:RFBurst:LEVel:RELative.

Video Trigger Delay

PXA commands and queries:

✔ :TRIGger[:SEquence]:VIDeo:DElay <time>

✔ :TRIGger[:SEquence]:VIDeo:DElay?

✔ :TRIGger[:SEquence]:VIDeo:DElay:STATe OFF|ON|0|1

✔ :TRIGger[:SEquence]:VIDeo:DElay:STATe?

In PSA, the Trigger Delay is global to all triggers, whereas in PXA, the delay can be set individually for each Trigger Source. When sent to PXA, the PSA :TRIGger[:SEquence]:DElay command affects the delay for the VIDEO, LINE, EXTERNAL1, and EXTERNAL2 triggers. The same is true for the ESA :TRIGger[:SEquence]:OFFSet command, when received by PXA.

Legacy Trigger Offset Command (for Backwards Compatibility Only):

✔ :TRIGger[:SEquence]:OFFSet <time>

✔ :TRIGger[:SEquence]:OFFSet?

✔ :TRIGger[:SEquence]:OFFSet:STATe OFF|ON|0|1

✔ :TRIGger[:SEquence]:OFFSet:STATe?

Trigger Commands

Video Trigger Level

Video Trigger Level

PXA commands and queries:

 :TRIGger[:SEquence]:VIDeo:LEVel <ampl>

 :TRIGger[:SEquence]:VIDeo:LEVel?

The following command and query are supported in PXA, for backward compatibility with PSA comms applications.

 :TRIGger[:SEquence]:IF:LEVel <ampl>

 :TRIGger[:SEquence]:IF:LEVel?

Video Trigger Slope

PXA commands and queries:

 :TRIGger[:SEquence]:VIDeo:SLOPe POSitive|NEGative

 :TRIGger[:SEquence]:VIDeo:SLOPe?

 :TRIGger[:SEquence]:SLOPe POSitive|NEGative

 :TRIGger[:SEquence]:SLOPe?

The following command and query are supported in PXA, for backward compatibility with PSA comms applications.

 :TRIGger[:SEquence]:IF:SLOPe NEGative|POSitive

 :TRIGger[:SEquence]:IF:SLOPe?

Note that, when sent to PXA, the PSA :TRIGger[:SEquence]:SLOPe command affects the slope for the VIDEO, LINE, EXTERNAL1, EXTERNAL2, and RFBURST triggers. See [“External 1, External 2 & Line Trigger Slope” on page 151](#) for more information.

Trigger Commands
Periodic Timer Sync Source

Periodic Timer Sync Source

PXA commands and queries:

 :TRIGger[:SEquence]:FRAMe:SYNC EXTerna11|EXTerna12|RFBurst|OFF

 :TRIGger[:SEquence]:FRAMe:SYNC?

For backwards compatibility, PXA supports the option `EXTerna1` and maps it to `EXTerna11`.

15 Preset Commands

Preset & Power-On Type

PXA command and query:

 :SYSTem:PON:TYPE MODE | USER | LAST

 :SYSTem:PON:TYPE?

PSA command, supported in PXA for backward compatibility:

 :SYSTem:PON:TYPE PRESet | LAST

PXA supports the PSA **PRESet** option for backwards compatibility. **PRESet** is mapped to **MODE**.

There is **no** Factory Preset in PXA. You can only preset the current mode, using either Mode Preset or User Preset. (See also “[Preset Type](#)” on page 159.)

PXA has no **Preset Type** key, and PSA’s **Power On** toggle key is replaced by the **System > Power On** softkey.

Compatibility issues for the three PXA Power On options are discussed in detail in the sections below.

Mode Preset (MODE)

In PXA, the settings and correction data under the **Input/Output** front-panel key (for example, Input Z Corr, Ext Amp Gain) are not part of any Mode, so they are not preset by a Mode Preset. They are preset using **Restore Input/Output Defaults**, **Restore System Defaults All** and they are recalled using User Preset (which performs a Recall State).

User Preset (USER)

In PXA, Power On User Preset causes the instrument to power up in the specified power-on mode (Measurement Application), **not** the last mode the instrument was in prior to shutdown. Power On User Preset also performs a User Preset on all modes.

Modes Affected by Preset

In PSA, performing a User Preset affects the state of the entire instrument. That is, all modes in the instrument are reset.

In PXA, User Preset only resets the state for the selected mode (Measurement Application). There is a User Preset file for each mode. If you want to reset **all** modes to their user preset file state, you must switch to each mode in turn, then perform a User Preset for each mode. The only way to affect the entire instrument is by using Restore System Defaults, which behaves essentially the same way it does in PSA.

Unlike the situation for PSA, User Preset in PXA can never cause a mode switch.

Preset Commands

Preset & Power-On Type

Data Affected by Preset

Performing a User Preset is equivalent to loading a state. In PSA, you can load any state, including User Preset, without affecting the trace data, limit lines or correction data, because that data is not part of the state.

Conversely, in PXA, the state includes all of this data, so whenever state is loaded, or User Preset is performed, all of the traces, limit lines and corrections are affected.

Last Preset (LAST)

In PSA, the instrument may be set to power up in its last running mode, with that mode in the **preset** state, by sending commands:

```
:SYSTem:PRESET:TYPE MODE; :SYSTem:PON:TYPE PRESET
```

PXA has no equivalent setting. The instrument may either be set to power up in its last running mode in its **last** state (via `:SYSTem:PON:TYPE LAST`), or to power up in the preset state of a specific mode (via `:SYSTem:PON:TYPE MODE; :SYSTem:PON:MODE <mode>`).

In PXA, correct operation of Power On Last depends on your shutting down the instrument using the Standby key or the `:SYSTem:PDOWn` command. This ensures that the last state of the modes is saved and can be recalled during a power up. (In PSA, you must set the instrument to the desired state at least two minutes prior to shutdown, to ensure that the last state is properly saved.)

Preset Type

PSA command and query:

 :SYSTem:PRESet:TYPE FACTory|MODE|USER

 :SYSTem:PRESet:TYPE?

In PXA, there is **no** "Preset Type" command (or menu), but, for backwards compatibility, the PSA command is accepted by PXA and ignored. "Factory Preset" has been replaced with Mode Preset, which only presets the currently selected mode, not the entire instrument.

In PXA, you can preset the entire instrument by using **System, Restore System Defaults All**, which behaves essentially the same way as **Restore System Defaults** in PSA.

PXA has a green **Mode Preset** front-panel key and a white **User Preset** front-panel key, each of which performs the named type of preset.

IEEE 488.2 Common Command



This command performs a Mode Preset in both PSA and PXA.

In PSA, *RST does not set the trigger mode to **Single**, but in PXA it does, for compliance with [IEEE Standard 488.2-1992](#).

In PXA, *RST does **not** perform a *CLS (clear the status bits and the error queue), to comply with [IEEE Standard 488.2-1992](#).

(In PSA, *RST performs the equivalent of the following sequence in PXA: :SYSTEM:PRESet; *CLS; :INITiate:CONTinuous OFF.)

16 System/Utility & IEEE 488.2 Common Commands

Align All Timeout



`:CALibration[:ALL]?`

Full alignment takes longer for PXA than for PSA.

Therefore, for PXA, the timeout period while waiting for alignment completion should be no less than 150 seconds.

System/Utility & IEEE 488.2 Common Commands

Auto Align

Auto Align

PXA command and query:

 :CALibration:AUTO ON|PARTial|OFF

 :CALibration:AUTO?

PSA command, supported in PXA for backward compatibility:

 :CALibration:AUTO ALERT

When this backward-compatibility command is sent to PXA, the **ALERT** parameter is mapped to **PARTial**.

In PSA, Auto Align **OFF** does not switch off all alignment, but is equivalent to **PARTial** in PXA.

In PXA, the option **OFF** switches alignment fully off. Thus, sending :CALibration:AUTO OFF (intended for PSA) to PXA may cause degraded performance. In such cases, the command should be changed to :CALibration:AUTO PARTial.

External Reference Output Control

PSA command and query:

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

 [:SENSe]:ROSCillator:OUTPut[:STATe]?

In PXA, these commands are **not** supported. The external reference output is always on.

System/Utility & IEEE 488.2 Common Commands

License Display

License Display

In ESA, the commands for displaying the Show Licenses screen, and querying its state, are:

 :SYSTem:CONFiGure:LKEY:STATe OFF|ON|0|1

 :SYSTem:CONFiGure:LKEY:STATe?

There are **no** equivalent commands in PXA for displaying the License Explorer.

RF Calibration Control

In PXA, the following command and query are provided only for ESA compatibility:

 :CALibration:SOURce:STATe OFF|ON|0|1

 :CALibration:SOURce:STATe?

The command, with the ON parameter, is aliased to the PXA command [SENSe]:FEED:AREFERENCE REF50. With the OFF parameter, it is aliased to PXA command [SENSe]:FEED:AREFERENCE OFF.

When the query CALibration:SOURce:STATe? is received, PXA returns 1 if any of the references is selected, or 0 if the Calibrator is off.

System/Utility & IEEE 488.2 Common Commands
Show System

Show System

The hardware statistics that are displayed in the PSA Show System screen have been moved to a dedicated Show Hardware Statistics screen, accessible via the **System > Service** menu.

System Error Handling



:SYSTem:ERRor[:NEXT]?

In PSA, the Repeat field shows the number of times the message has repeated since the last time the error queue was cleared. In PXA, the Repeat field shows the number of times the error has repeated since the last intervening error, thus the displayed counts may differ between the two instruments, even for identical signal conditions

In PXA, all errors are reported through the Message or Status lines and are logged to the event queue. They never appear as text in the graticule area, and they are never displayed in the settings panel at the top of the screen.

As a consequence of the above, you can only see one status condition (the most recently generated) without looking at the queue.

IMPORTANT

In general there is no specified or guaranteed backwards compatibility between the error numbers in X-Series and those of earlier products. Thus any error, event, and status processing code will probably need to be rewritten to work with PXA.

In PSA, some conditions report as errors, while others simply turn on status bits. Conditions that report as errors often report repeatedly, for as long as the condition exists. In PXA, all conditions report as start and stop events. Consequently, PSA-targeted software that repeatedly queries for a condition error until it stops reporting must be modified for PXA.

System/Utility & IEEE 488.2 Common Commands
System Language Query

System Language Query



:SYSTem:LANGuage?

PXA supports this query **only** when the N9061A Remote Language Compatibility Measurement Application is selected.

For PXA Measurement Applications that do not support this query, nothing is returned.

System Power-On Type Options

See [“Preset & Power-On Type”](#) on page 157.

Recall

In PSA, it is possible to load a state without affecting the trace data, limit lines or correction data. Similarly (since User Preset actually loads a predetermined state), it is possible to do a User Preset without affecting the trace data, limit lines or correction data.

In PXA, “state” always includes all of this data, so whenever state is loaded, all of the traces, limit lines and corrections are affected.

NOTE There is no compatibility between state files for PXA and PSA; for example, you cannot recall a state file from PSA into PXA.

Load from File

PXA command:

 :MMEMory:LOAD:STATe <filename>

PSA command, supported in PXA for backward compatibility:

 :MMEMory:LOAD:STATe 1,<filename>

When this backward-compatibility command is sent to PXA, the parameter 1 is ignored.

Amplitude Correction

PXA command:

 :MMEMory:LOAD:CORRection 1|2|3|4|5|6, <filename>

PSA command, supported in PXA for backward compatibility:

 :MMEMory:LOAD:CORRection ANTenna|CABLe|OTHer|USER, <filename>

When this backward-compatibility command is sent to PXA, the parameters are mapped as follows:

ANTenna	1
CABLe	2
OTHer	3
USER	4

Default Path & File Name

See “[Default Path & File Name](#)” on page 171.

Save

Save to File

PXA command:

 :MMEMory:STORe:STATe <filename>

PSA command, supported in PXA for backward compatibility:

 :MMEMory:STORe:STATe 1,<filename>

When this backward-compatibility command is sent to PXA, the parameter 1 is ignored. The command is sequential.

Amplitude Correction

PXA command:

 :MMEMory:STORe:CORRection 1|2|3|4|5|6, <filename>

PSA command, supported in PXA for backward compatibility:

 :MMEMory:STORe:CORRection ANTenna|CABLe|OTHer|USER, <filename>

When this backward-compatibility command is sent to PXA, the parameters are mapped as follows:

ANTenna	1
CABLe	2
OTHer	3
USER	4

Default Path & File Name

PSA uses C:\ as its default folder for storing setting files.

PXA uses <MyDocuments>\System as its default folder, but note that the actual path represented by <MyDocuments> may differ according to system configuration and user name. You can obtain the actual path by using the query `MMEMory:CDIRectory?`.

PSA uses `REG0n.STA` as the default name for Register State files, whereas PXA uses `Register00n.state`.

Frequency Reference Selection

PXA command and query:

 [:SENSe]:ROSCillator:SOURce:TYPE INTernal | EXTernal | SENSe

 [:SENSe]:ROSCillator:SOURce:TYPE?

In PSA, the Freq Ref In setting is not saved as part of the instrument state, whereas in PXA this setting is saved as part of the state.

PSA command and query, both of which are accepted by PXA:

 [:SENSe]:ROSCillator:SOURce INTernal | EXTernal

PXA maps this command to [:SENSe]:ROSCillator:SOURce:TYPE INTernal | EXTernal.

 [:SENSe]:ROSCillator:SOURce?

This query returns the actual routing (internal or external), and is backwards-compatible with both instruments.

Timebase DAC Calibration

In PXA, the following are used to adjust or query the timebase:

 :CALibration:FREQuency:REFeRence:FINE <integer>

 :CALibration:FREQuency:REFeRence:FINE?

For ESA compatibility, the keyword COARse may be substituted for FINE, so the following aliases may be used:

 :CALibration:FREQuency:REFeRence:COARse <integer>

 :CALibration:FREQuency:REFeRence:COARse?

IEEE 488.2 Common Commands

Clear Status



In general the status bits used in the PXA status system are backwards compatible with ESA and PSA.

However, note that all conditions generate events that are sent to the event log. Some conditions also modify status bits. See also [Service Request Enable](#) below.

Operation Complete



In PSA, commands such as, *OPC/*OPC?, *WAI, *RST are global. In PXA, each command is evaluated on a per-channel basis. That is, the various rear panel remote ports and the front panel I/O are all considered separately. Only the functionality initiated on the port where the *OPC command was sent is considered for its operation.

In PXA, *OPC holds off until **all** overlapping commands are completed.

In PSA, *OPC does **not** wait for completion of all processes, and holds off only until the following bits are cleared in the STATUS:OPERation register:

- CALibrating
- SWEeping
- Waiting for TRIGger
- MEASuring (but not in all Modes)

Preset



See "[IEEE 488.2 Common Command](#)" on page 160.

Service Request Enable



Unlike PSA, PXA retains the *SRE mask setting following a *CLS.

Prior to software revision 07, PXA clears *SRE mask with *CLS.

17 Miscellaneous Commands

External Mixer Presets

PXA command and query:



[:SENSe] :MIXer :BAND

A|Q|U|V|W|NA|ND|NE|NF|NG|NJ|NK|NQ|NU|NV|NW|NY|NEXT|DD|DF|DG|DJ|DK|DQ|DV|DW|
DY|DEXT|MA|ME|MU|MCOAX



[:SENSe] :MIXer :BAND?

PSA command and query:



[:SENSe] :MIXer :BAND K|A|Q|U|V|E|W|F|D|G|Y|J|USER



[:SENSe] :MIXer :BAND?

This command is used in PSA to select the mixer band.

In PXA, only the options A, Q, U, V, and W are supported, and they preset the instrument to match the corresponding Keysight 11970 mixer. Options D, E, F, G, J, K, Y, which are accepted in PSA, return an error if sent to PXA. If you are using a mixer in one of these bands, you should study the tables of presets and choose the appropriate preset to match your application. PXA does not accept the USER option, because its control model for mixer customization is substantially different.

Miscellaneous Commands
External Mixer Harmonic Value

External Mixer Harmonic Value

PSA and PXA command and query:

 [:SENSe]:MIXer:HARMonic <integer>

 [:SENSe]:MIXer:HARMonic?

PSA-only command and query:

 [:SENSe]:MIXer:HARMonic:AUTO OFF|ON|0|1

 [:SENSe]:MIXer:HARMonic:AUTO?

PXA does **not** support the AUTO command or query.

A: References

1. **IEEE Standard 488.2-1992**
IEEE Standard Codes, Formats, Protocols, and Common Commands for Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation
May be downloaded in Acrobat (PDF) format from:
ieeexplore.ieee.org/iel1/2839/5581/00213762.pdf?arnumber=213762

References