



# SignalShark® Series

---

## Command Reference Guide

Narda Safety Test Solutions GmbH  
Sandwiesenstraße 7  
72793 Pfullingen, Germany  
© 2022

® Names and Logo are registered trademarks of Narda Safety Test  
Solutions GmbH - Trade names are trademarks of the owners.

Document no.: 3300/98.24  
Version: 2022-01, (Firmware V1.7.5)  
Previous version: 2020-07-30

Subject to change.  
Our normal guarantee and delivery terms apply.  
Printed in Germany



# 1 Contents

1	Contents .....	4
2	General Information .....	22
	SignalShark Network Setup .....	22
3	Narda SCPI basic principles .....	24
	Narda SCPI information .....	24
	Various parsers .....	24
	Syntax convention.....	24
	Basic sequence for a remote-controlled measurement .....	25
	Defined start state .....	25
	End character .....	25
	Commands .....	25
	Queries .....	25
	Parameter corrections for Narda devices .....	25
	SCPI error queue .....	26
	SCPI and Narda error codes .....	26
	Task States (RUN, STOP, HOLD).....	27
	Measurement parameters and evaluation parameters .....	27
	Synchronizing several commands .....	27
	Concatenating commands and queries using ; .....	28
	Remote Logfile .....	28
	RUN:CONTInuous Commands .....	28
	RUN:SINGle Commands .....	28
	DATA:ALL? commands .....	28
	DATA:UPDate Commands .....	30
	Commands with long Timeouts .....	31
	Progress query for long Timeouts.....	31
	GUI concept as template for the Narda parser .....	33
4	Narda SCPI Data types .....	34
5	Stream Communication.....	39
	General information .....	39
	General Stream Header .....	41
6	Narda Audio Stream Context .....	42
7	Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark .....	43
	Spectrum Data Packet Class – Overview .....	43

	Spectrum Context Packet Class – Overview.....	44
	Spectrum Data Packet Class – Details .....	46
	Spectrum Context Packet Class - Details .....	48
8	Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark.....	55
	IF Data Packet Class – Overview.....	55
	IF Context Packet Class – Overview .....	56
	IF Data Packet Class – Details.....	58
	IF Context Packet Class - Details.....	60
	Glossary.....	64
9	Command Groups.....	65
10	Command Descriptions.....	87
	Narda Command Reference - IEEE488 .....	87
	*IDN? .....	87
	*OPC?.....	87
	*RST .....	87
	*WAI.....	87
	Narda Command Reference - SCPI-99.....	89
	ABORT.....	89
	SYSTem:ERRor[:NEXT]? .....	89
	SYSTem:ERRor:ALL? .....	89
	SYSTem:ERRor:CLEar[:ALL].....	89
	SYSTem:ERRor:CODE[:NEXT]? .....	90
	SYSTem:ERRor:CODE:ALL?.....	90
	SYSTem:ERRor:COUNt? .....	90
	SYSTem:ERRor:LIST? .....	90
	Narda Command Reference - DEVICE .....	92
	DEvice:APPLication:DATE? .....	92
	DEvice:APPLication:VERSion? .....	92
	DEvice:BIOS:VERSion? .....	92
	DEvice:BOARd:CONTroller:BOOTloader:VERSion? .....	92
	DEvice:BOARd:CONTroller:FIRMware:VERSion? .....	93
	DEvice:FPGA:BITStream:VERSion? .....	93
	DEvice:FPGA:PCIE:DRIVER:VERSion? .....	93
	DEvice:OPERation:SYSTem:VERSion? .....	94
	DEvice:PART:NUMBer? .....	94
	DEvice:PRODuct:NAME? .....	94

# Command Reference Guide

## Contents

---

DEvice:SErIal:NUMBer?	94
Narda Command Reference - SETTings	96
SETTings:ADJust:EQUalizer	96
SETTings:DATE?	96
SETTings:FAST:SCANs:ONLY	96
SETTings:FAST:SCANs:ONLY?	96
SETTings:GNSS:SOURce	97
SETTings:GNSS:SOURce?	97
SETTings:MAGNetic:DECLination	97
SETTings:MAGNetic:DECLination?	97
SETTings:PPS:SOURce	98
SETTings:PPS:SOURce?	98
SETTings:REFeRence:FREQuency:SOURce	98
SETTings:REFeRence:FREQuency:SOURce?	99
SETTings:TIME?	99
SETTings:TIME:SYNC:NOW	99
SETTings:TSTamp:SYNC:MODE	99
SETTings:TSTamp:SYNC:MODE?	100
SETTings:TSTamp:SYNC:NOW	100
Narda Command Reference - SENSE	101
[SENSe:]ADC:OOR?	101
[SENSe:]ANTenna:AUTO:DATA?	101
[SENSe:]ANTenna:AUTO:INFO?	101
[SENSe:]ANTenna:CHECKsum?	102
[SENSe:]ANTenna:HANDle?	102
[SENSe:]ANTenna:ISOTropic:MANual	102
[SENSe:]ANTenna:ISOTropic:MANual?	103
[SENSe:]ANTenna:ISOTropic:MANual:AXIS	103
[SENSe:]ANTenna:ISOTropic:MANual:AXIS?	103
[SENSe:]ANTenna:MANual	104
[SENSe:]ANTenna:MANual?	104
[SENSe:]ANTenna:MANual:LIST?	104
[SENSe:]ANTenna:POLarization?	105
[SENSe:]ANTenna:TYPE?	105
[SENSe:]ATTenuator	105
[SENSe:]ATTenuator?	105

[SENSe:]ATTenuator:LIST? .....	106
[SENSe:]CABLe:AUTO:DATA? .....	106
[SENSe:]CABLe:AUTO:INFO? .....	107
[SENSe:]CABLe:CHECksum? .....	107
[SENSe:]CABLe:MANual .....	107
[SENSe:]CABLe:MANual? .....	108
[SENSe:]CABLe:MANual:LIST? .....	108
[SENSe:]CABLe:TYPE? .....	108
[SENSe:]COMPass:DATA? .....	109
[SENSe:]EXTErnal:DEVIce .....	109
[SENSe:]EXTErnal:DEVIce? .....	109
[SENSe:]EXTErnal:DEVIce:MANual .....	110
[SENSe:]EXTErnal:DEVIce:MANual? .....	110
[SENSe:]EXTErnal:DEVIce:MANual:LIST? .....	110
[SENSe:]EXTErnal:DEVIce:SWITChable? .....	111
[SENSe:]FREQuency:RANGe? .....	111
[SENSe:]GNSS:DATA? .....	111
[SENSe:]HOLD .....	112
[SENSe:]HOLD? .....	112
[SENSe:]INPut .....	113
[SENSe:]INPut? .....	113
[SENSe:]PREamp .....	114
[SENSe:]PREamp? .....	114
[SENSe:]REFerence:LEVel .....	114
[SENSe:]REFerence:LEVel? .....	114
[SENSe:]REFerence:LEVel:LIST? .....	115
[SENSe:]REFerence:LEVel:OFFSet .....	115
[SENSe:]REFerence:LEVel:OFFSet? .....	115
[SENSe:]REFerence:LEVel:OFFSet:ENABle .....	116
[SENSe:]REFerence:LEVel:OFFSet:ENABle? .....	116
[SENSe:]REFerence:LEVel:OFFSet:LIST? .....	116
[SENSe:]RUN:CONTInuous .....	116
[SENSe:]RUN:CONTInuous? .....	117
[SENSe:]RUN:CONTInuous:RESet .....	117
[SENSe:]RUN:CONTInuous:RESet? .....	118
[SENSe:]RUN:SINGle .....	118

# Command Reference Guide

## Contents

---

[SENSe:]RUN:SINGLE?	119
[SENSe:]STOP	119
[SENSe:]TSTamp:SYNC:DEVIation?	119
[SENSe:]TSTamp:SYNC:FINE?	120
[SENSe:]TSTamp:SYNC:STATe?	120
Narda Command Reference - DISPlay	121
DISPlay:LEVelmeter:LMAX	121
DISPlay:LEVelmeter:LMAX?	121
DISPlay:LEVelmeter:LRANge	121
DISPlay:LEVelmeter:LRANge?	121
DISPlay:MAP:CENTer	122
DISPlay:MAP:CENTer:POSition	122
DISPlay:MAP:CENTer:POSition?	122
DISPlay:MAP:LIST?	122
DISPlay:MAP:SElect	123
DISPlay:MAP:SElect?	123
DISPlay:MAP:ZOOM:LEVel	123
DISPlay:MAP:ZOOM:LEVel?	124
DISPlay:PEAKtable:SHOW:TRANsmitter	124
DISPlay:PEAKtable:SHOW:TRANsmitter?	124
DISPlay:PEAKtable:SORT	124
DISPlay:PEAKtable:SORT?	125
DISPlay:PERStence:LMAX	125
DISPlay:PERStence:LMAX?	125
DISPlay:PERStence:LRANge	125
DISPlay:PERStence:LRANge?	126
DISPlay:SPECTrum:LMAX	126
DISPlay:SPECTrum:LMAX?	126
DISPlay:SPECTrum:LRANge	127
DISPlay:SPECTrum:LRANge?	127
DISPlay:UNIT	127
DISPlay:UNIT?	127
DISPlay:UNIT:LIST?	128
Narda Command Reference - DLOGger	129
DLOGger:SAVE:LOCalization	129
DLOGger:WORKing:DIRectory	129



DLOGger:WORKing:DIRectory? .....	129
DLOGger:CONFig:TASK:SAVE .....	129
DLOGger:CONFig:TASK:SAVE? .....	130
DLOGger:DIRectory:ADD? .....	130
DLOGger:DIRectory:DELeTe .....	130
DLOGger:DIRectory:DELeTe:ALL .....	131
DLOGger:DIRectory:LIST? .....	131
DLOGger:RECall:DATaset .....	131
DLOGger:RECall:DATaset? .....	131
DLOGger:RECall:DATaset:CLOSe .....	132
DLOGger:RECall:ENABled .....	132
DLOGger:RECall:ENABled? .....	132
DLOGger:SAVE:DATaset .....	133
DLOGger:SAVE:EXTeRnal:BEARing .....	133
DLOGger:SAVE:SCReenshot .....	133
Narda Command Reference - SYSTem .....	135
SYSTem:AUDio:MUTE .....	135
SYSTem:AUDio:MUTE? .....	135
SYSTem:AUDio:VOLume .....	135
SYSTem:AUDio:VOLume? .....	135
SYSTem:COMMand:FILTer? .....	136
SYSTem:COMMand:PROGress? .....	136
SYSTem:MEMory:STATus? .....	136
SYSTem:REMOte:COMMand:LIST? .....	137
SYSTem:REMOte:DISPlay .....	137
SYSTem:REMOte:DISPlay? .....	137
SYSTem:REMOte:LOG .....	138
SYSTem:REMOte:LOG? .....	138
SYSTem:REMOte:LOG:CONFig .....	138
SYSTem:REMOte:LOG:CONFig? .....	139
SYSTem:REMOte:SLEep? .....	139
SYSTem:REMOte:TIMeout .....	140
SYSTem:REMOte:TIMeout? .....	140
SYSTem:REMOte:TIMeout:RESet .....	140
SYSTem:SHUTdown .....	140
SYSTem:STARtup:DATE? .....	141

# Command Reference Guide

## Contents

---

SYSTem:STARtup:SEConds? .....	141
SYSTem:STARtup:TIME? .....	141
Narda Command Reference - NETWork .....	143
NETWork:MAC:ADDReSS? .....	143
Narda Command Reference - TASK .....	144
TASK:ADD? .....	144
TASK:DELeTe .....	144
TASK:DELeTe:ALL .....	144
TASK:LIST? .....	144
TASK:MOVE .....	145
TASK:NEW? .....	145
TASK:REName .....	145
TASK:REPLace? .....	146
TASK:SElect .....	146
TASK:SElect? .....	146
TASK:STATe? .....	147
Narda Command Reference - VIEW .....	148
VIEW:ADD? .....	148
VIEW:DELeTe .....	148
VIEW:LIST? .....	148
VIEW:REPLace .....	149
VIEW:SElect .....	149
VIEW:SElect? .....	149
VIEW:SIZE:MAXimize .....	149
VIEW:SIZE:REStore .....	150
Narda Command Reference - SPECTrum .....	151
SPECTrum:DATA:ALL? .....	151
SPECTrum:DATA:COUNt? .....	151
SPECTrum:DATA:FREQuency:STARt? .....	152
SPECTrum:DATA:FREQuency:STEP? .....	152
SPECTrum:DATA:LEVel? .....	152
SPECTrum:DATA:OVERdriven? .....	153
SPECTrum:DATA:REALtime? .....	153
SPECTrum:DATA:UPDate? .....	153
SPECTrum:FREQuency:CENTer .....	154
SPECTrum:FREQuency:CENTer? .....	154

SPECTrum:FREQuency:CENTer:STEP .....	155
SPECTrum:FREQuency:CENTer:STEP? .....	155
SPECTrum:FREQuency:CENTer:TUNE:COUPling .....	155
SPECTrum:FREQuency:CENTer:TUNE:COUPling? .....	155
SPECTrum:FREQuency:ENTRy:MODE .....	156
SPECTrum:FREQuency:ENTRy:MODE? .....	156
SPECTrum:FREQuency:SPAN .....	156
SPECTrum:FREQuency:SPAN? .....	157
SPECTrum:FREQuency:STARt .....	157
SPECTrum:FREQuency:STARt? .....	157
SPECTrum:FREQuency:STOP .....	157
SPECTrum:FREQuency:STOP? .....	158
SPECTrum:MEASurement:TIME .....	158
SPECTrum:MEASurement:TIME? .....	158
SPECTrum:MEASurement:TIME:ENTRy:MODE .....	158
SPECTrum:MEASurement:TIME:ENTRy:MODE? .....	159
SPECTrum:RBW .....	159
SPECTrum:RBW? .....	159
SPECTrum:RBW:AUTO .....	160
SPECTrum:RBW:AUTO? .....	160
SPECTrum:RBW:ENTRy:MODE .....	160
SPECTrum:RBW:ENTRy:MODE? .....	160
SPECTrum:RBW:FILTer:TYPE .....	161
SPECTrum:RBW:FILTer:TYPE? .....	161
SPECTrum:RBW:LIST? .....	161
SPECTrum:SCAN:COUNt .....	162
SPECTrum:SCAN:COUNt? .....	162
SPECTrum:SCAN:NUMBer? .....	162
SPECTrum:TRACe:DETEctor<Number> .....	162
SPECTrum:TRACe:DETEctor<Number>? .....	163
SPECTrum:TRACe:ENABle .....	163
SPECTrum:TRACe:ENABle? .....	163
SPECTrum:TRACe:INFinite .....	164
SPECTrum:TRACe:INFinite? .....	164
SPECTrum:TRACe:LIST? .....	164
SPECTrum:TSTamp? .....	165

# Command Reference Guide

## Contents

---

Narda Command Reference - MARKer .....	166
MARKer:FXD:FREQuency .....	166
MARKer:FXD:FREQuency? .....	166
MARKer:FXD:TIME .....	166
MARKer:FXD:TIME? .....	166
MARKer:FXD:VALue .....	167
MARKer:FXD:VALue? .....	167
MARKer:SPECTrum:DATA:ALL? .....	167
MARKer<Index>:SPECTrum:DATA:FREQuency? .....	168
MARKer<Index>:SPECTrum:DATA:LEVel? .....	169
MARKer<Index>:SPECTrum:DATA:PEAK:STATe? .....	169
MARKer<Index>:SPECTrum:DATA:TIME? .....	169
MARKer<Index>:SPECTrum:ENABLE .....	170
MARKer<Index>:SPECTrum:ENABLE? .....	170
MARKer<Index>:SPECTrum:FREQuency .....	171
MARKer<Index>:SPECTrum:FREQuency? .....	171
MARKer<Index>:SPECTrum:FREQuency:LINK .....	171
MARKer<Index>:SPECTrum:FREQuency:LINK? .....	172
MARKer<Index>:SPECTrum:FREQuency:LINK:OFFSet .....	172
MARKer<Index>:SPECTrum:FREQuency:LINK:OFFSet? .....	173
MARKer<Index>:SPECTrum:FUNCTion .....	173
MARKer<Index>:SPECTrum:FUNCTion? .....	174
MARKer:SPECTrum:FUNCTion:CPOWer:CBW .....	174
MARKer:SPECTrum:FUNCTion:CPOWer:CBW? .....	174
MARKer<Index>:SPECTrum:FUNCTion:CPOWer:DATA? .....	174
MARKer<Index>:SPECTrum:FUNCTion:NOISe:DATA? .....	175
MARKer:SPECTrum:FUNCTion:NOISe:NBW .....	175
MARKer:SPECTrum:FUNCTion:NOISe:NBW? .....	176
MARKer:SPECTrum:FUNCTion:NOISe:NBW:AUTO .....	176
MARKer:SPECTrum:FUNCTion:NOISe:NBW:AUTO? .....	176
MARKer<Index>:SPECTrum:FUNCTion:OCBW:DATA? .....	176
MARKer:SPECTrum:FUNCTion:OCBW:MODE .....	177
MARKer:SPECTrum:FUNCTion:OCBW:MODE? .....	177
MARKer:SPECTrum:FUNCTion:OCBW:NTRials .....	177
MARKer:SPECTrum:FUNCTion:OCBW:NTRials? .....	178
MARKer:SPECTrum:FUNCTion:OCBW:PERCent .....	178

MARKer:SPECTrum:FUNCTion:OCBW:PERCent?	178
MARKer:SPECTrum:FUNCTion:OCBW:THReshold	179
MARKer:SPECTrum:FUNCTion:OCBW:THReshold?	179
MARKer:SPECTrum:FUNCTion:OCBW:XDB	179
MARKer:SPECTrum:FUNCTion:OCBW:XDB?	180
MARKer<Index>:SPECTrum:FUNCTion:TRANsmitter:DATA?	180
MARKer:SPECTrum:LIST?	180
MARKer<Index>:SPECTrum:PEAK	181
MARKer<Index>:SPECTrum:PEAK:LEFT	181
MARKer<Index>:SPECTrum:PEAK:LOWer	181
MARKer<Index>:SPECTrum:PEAK:NEXT	182
MARKer<Index>:SPECTrum:PEAK:RIGHT	182
MARKer<Index>:SPECTrum:PEAK:UPPer	182
MARKer<Index>:SPECTrum:REFerence	182
MARKer<Index>:SPECTrum:REFerence?	183
MARKer:SPECTrum:SEARch:AUTO:PEAK:ENABle	183
MARKer:SPECTrum:SEARch:AUTO:PEAK:ENABle?	183
MARKer:SPECTrum:SEARch:FREQuency:LOWer	184
MARKer:SPECTrum:SEARch:FREQuency:LOWer?	184
MARKer:SPECTrum:SEARch:FREQuency:UPPer	184
MARKer:SPECTrum:SEARch:FREQuency:UPPer?	185
MARKer:SPECTrum:SEARch:LIMits:ENABle	185
MARKer:SPECTrum:SEARch:LIMits:ENABle?	185
MARKer:SPECTrum:SEARch:LOEXclude:ENABle	185
MARKer:SPECTrum:SEARch:LOEXclude:ENABle?	186
MARKer:SPECTrum:SEARch:PEAK:EXCursion	186
MARKer:SPECTrum:SEARch:PEAK:EXCursion?	186
MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABle	186
MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABle?	187
MARKer:SPECTrum:SEARch:SGRam:RANGe	187
MARKer:SPECTrum:SEARch:SGRam:RANGe?	187
MARKer:SPECTrum:SEARch:THReshold	188
MARKer:SPECTrum:SEARch:THReshold?	188
MARKer:SPECTrum:SEARch:TRACK:PEAKs:ENABle	188
MARKer:SPECTrum:SEARch:TRACK:PEAKs:ENABle?	189
MARKer<Index>:SPECTrum:TIME	189

# Command Reference Guide

## Contents

---

MARKer<Index>:SPECTrum:TIME?	189
MARKer<Index>:SPECTrum:TIME:LINK	190
MARKer<Index>:SPECTrum:TIME:LINK?	190
MARKer<Index>:SPECTrum:TIME:LINK:OFFSet	190
MARKer<Index>:SPECTrum:TIME:LINK:OFFSet?	191
MARKer<Index>:SPECTrum:TRACe	191
MARKer<Index>:SPECTrum:TRACe?	191
MARKer<Index>:SPECTrum:TYPE	192
MARKer<Index>:SPECTrum:TYPE?	192
Narda Command Reference - LEVelmeter	194
LEVelmeter:CBW	194
LEVelmeter:CBW?	194
LEVelmeter:CBW:FILTer:TYPE	194
LEVelmeter:CBW:FILTer:TYPE?	194
LEVelmeter:CBW:LIST?	195
LEVelmeter:CBW:OVERsampling	195
LEVelmeter:CBW:OVERsampling?	195
LEVelmeter:DATA:AFC?	196
LEVelmeter:DATA:ALL?	196
LEVelmeter:DATA:DETEctor<Number>?	198
LEVelmeter:DATA:DETEctor:MODulation?	198
LEVelmeter:DATA:UPDate?	198
LEVelmeter:DETEctor<Number>	199
LEVelmeter:DETEctor<Number>?	200
LEVelmeter:DETEctor:INFinite	200
LEVelmeter:DETEctor:INFinite?	200
LEVelmeter:DETEctor:MODulation	200
LEVelmeter:DETEctor:MODulation?	201
LEVelmeter:FREQuency:TUNE	201
LEVelmeter:FREQuency:TUNE?	201
LEVelmeter:FREQuency:TUNE:CENTer:COUPling	202
LEVelmeter:FREQuency:TUNE:CENTer:COUPling?	202
LEVelmeter:FREQuency:TUNE:STEP	202
LEVelmeter:FREQuency:TUNE:STEP?	202
LEVelmeter:MEASurement:TIME	203
LEVelmeter:MEASurement:TIME?	203

LEVelmeter:POST:AVG .....	203
LEVelmeter:POST:AVG? .....	204
LEVelmeter:PRE:AVG .....	204
LEVelmeter:PRE:AVG? .....	204
LEVelmeter:SCAN:COUNt .....	204
LEVelmeter:SCAN:COUNt? .....	205
LEVelmeter:SCAN:NUMBer? .....	205
LEVelmeter:TSTamp? .....	205
Narda Command Reference - PEAKtable .....	206
PEAKtable:DATA:ALL? .....	206
PEAKtable:DATA:COUNt? .....	206
PEAKtable:DATA:FREQuency? .....	207
PEAKtable:DATA:LEVel? .....	207
PEAKtable:DATA:SCAN:NUMBer? .....	207
PEAKtable:DATA:TSTamp? .....	208
PEAKtable:DATA:UPDate? .....	208
PEAKtable:SCAN:NUMBer? .....	209
PEAKtable:TRACe .....	209
PEAKtable:TRACe? .....	209
PEAKtable:TSTamp? .....	210
Narda Command Reference - STReam .....	211
STReam:ADD? .....	211
STReam:CONNection:ADDReSS .....	211
STReam:CONNection:ADDReSS? .....	212
STReam:CONNection:CLOSe .....	212
STReam:CONNection:IDN .....	212
STReam:CONNection:IDN? .....	212
STReam:CONNection:OPEN .....	213
STReam:CONNection:PORT .....	213
STReam:CONNection:PORT? .....	213
STReam:CONNection:STATe? .....	213
STReam:CONNection:TYPE .....	214
STReam:CONNection:TYPE? .....	214
STReam:DELeTe .....	214
STReam:LIST? .....	214
STReam:SELeCt .....	215

# Command Reference Guide

## Contents

---

STReam:SElect? .....	215
Narda Command Reference - IQSTream .....	216
IQSTream:ANTenna:FACTor? .....	216
IQSTream:CBW .....	216
IQSTream:CBW? .....	216
IQSTream:CBW:LIST? .....	216
IQSTream:FREQuency:TUNE .....	217
IQSTream:FREQuency:TUNE? .....	217
IQSTream:OVERsampling .....	217
IQSTream:OVERsampling? .....	218
IQSTream:PAYLoad:FORMat .....	218
IQSTream:PAYLoad:FORMat? .....	218
IQSTream:PAYLoad:LIMit .....	219
IQSTream:PAYLoad:LIMit? .....	219
IQSTream:PAYLoad:LIMit:LIST? .....	219
Narda Command Reference - BEARing .....	220
BEARing:AZIMuth:CORRection .....	220
BEARing:AZIMuth:CORRection? .....	220
BEARing:CBW .....	220
BEARing:CBW? .....	220
BEARing:CBW:LIST? .....	221
BEARing:CYCLe:TIME? .....	221
BEARing:DATA:ALL? .....	221
BEARing:DATA:AZIMuth? .....	223
BEARing:DATA:DETEctor? .....	223
BEARing:DATA:DFQuality? .....	223
BEARing:DATA:ELEVation? .....	223
BEARing:DATA:SCAN:NUMBer? .....	224
BEARing:DATA:TSTamp? .....	224
BEARing:DATA:UPDate? .....	224
BEARing:DATA:VALid? .....	225
BEARing:DATA:VALid:ENABLE .....	225
BEARing:DATA:VALid:ENABLE? .....	226
BEARing:DFCorrection:LIST? .....	226
BEARing:DFCorrection:SElect .....	226
BEARing:DFCorrection:SElect? .....	227



BEARing:DFSQuelch .....	227
BEARing:DFSQuelch? .....	227
BEARing:DFSQuelch:ENABLE.....	228
BEARing:DFSQuelch:ENABLE? .....	228
BEARing:FREQuency:TUNE.....	228
BEARing:FREQuency:TUNE?.....	229
BEARing:FREQuency:TUNE:STEP .....	229
BEARing:FREQuency:TUNE:STEP? .....	229
BEARing:MEASurement:TIME .....	229
BEARing:MEASurement:TIME? .....	230
BEARing:MIN:DFQuality .....	230
BEARing:MIN:DFQuality? .....	230
BEARing:MIN:DFQuality:ENABLE .....	231
BEARing:MIN:DFQuality:ENABLE? .....	231
BEARing:MIN:STABility .....	231
BEARing:MIN:STABility? .....	232
BEARing:MIN:STABility:ENABLE .....	232
BEARing:MIN:STABility:ENABLE? .....	232
BEARing:NORTH:REFerence.....	232
BEARing:NORTH:REFerence?.....	233
BEARing:POST:AVG.....	233
BEARing:POST:AVG?.....	233
BEARing:REFerence:MARK:DIRection .....	234
BEARing:REFerence:MARK:DIRection? .....	234
BEARing:SCAN:COUNt .....	234
BEARing:SCAN:COUNt? .....	234
BEARing:SCAN:NUMBer? .....	235
BEARing:SCAN:TIME? .....	235
BEARing:TSTamp? .....	235
Narda Command Reference - DEMod .....	237
DEMod:CBW .....	237
DEMod:CBW? .....	237
DEMod:CBW:FILTer:TYPE .....	237
DEMod:CBW:FILTer:TYPE? .....	237
DEMod:CBW:LIST?.....	238
DEMod:CBW:OVERsampling.....	238

# Command Reference Guide

## Contents

---

DEMod:CBW:OVERsampling? .....	238
DEMod:FREQuency:TUNE .....	239
DEMod:FREQuency:TUNE? .....	239
DEMod:FREQuency:TUNE:AFC .....	239
DEMod:FREQuency:TUNE:AFC? .....	239
DEMod:FREQuency:TUNE:BFO .....	240
DEMod:FREQuency:TUNE:BFO? .....	240
DEMod:MUTE .....	240
DEMod:MUTE? .....	241
DEMod:SQUelch .....	241
DEMod:SQUelch? .....	241
DEMod:SQUelch:ENABle .....	241
DEMod:SQUelch:ENABle? .....	242
DEMod:TYPE .....	242
DEMod:TYPE? .....	242
DEMod:TYPE:ENABle .....	242
DEMod:TYPE:ENABle? .....	243
DEMod:VOLume .....	243
DEMod:VOLume? .....	243
DEMod:VOLume:AGC .....	244
DEMod:VOLume:AGC? .....	244
Narda Command Reference - SGRam .....	245
SGRam:DATA:ALL? .....	245
SGRam:DATA:FRAMe:COUNT? .....	246
SGRam:DATA:UPDate? .....	246
SGRam:DETEctor<Number> .....	247
SGRam:DETEctor<Number>? .....	247
SGRam:DETEctor:VISible .....	248
SGRam:DETEctor:VISible? .....	248
SGRam:FRAMe:COUNT .....	248
SGRam:FRAMe:COUNT? .....	249
SGRam:SCAN:NUMBer? .....	249
Narda Command Reference - MAP .....	250
MAP:LOCalization:AREA .....	250
MAP:LOCalization:AREA? .....	250
MAP:LOCalization:BEARing:ERRor .....	250

MAP:LOCalization:BEARing:ERRor?	251
MAP:LOCalization:DATA?	251
MAP:LOCalization:FREQuency:TUNE:LIST?	252
MAP:LOCalization:FREQuency:TUNE:SElect	252
MAP:LOCalization:FREQuency:TUNE:SElect?	252
MAP:LOCalization:LOS:PROBability	252
MAP:LOCalization:LOS:PROBability?	253
MAP:LOCalization:MIN:DFQuality	253
MAP:LOCalization:MIN:DFQuality?	253
MAP:LOCalization:MIN:DFQuality:ENABLE	254
MAP:LOCalization:MIN:DFQuality:ENABLE?	254
MAP:LOCalization:MIN:SPEed	254
MAP:LOCalization:MIN:SPEed?	254
MAP:LOCalization:MIN:SPEed:ENABLE	255
MAP:LOCalization:MIN:SPEed:ENABLE?	255
MAP:LOCalization:MIN:SPEed:LIST?	255
MAP:LOCalization:RECOrd:BEARing	256
MAP:LOCalization:RECOrd:BEARing?	256
MAP:LOCalization:RECOrd:LIST?	256
MAP:LOCalization:RECOrd:SElect	256
MAP:LOCalization:RECOrd:SElect?	257
MAP:LOCalization:RESolution	257
MAP:LOCalization:RESolution?	257
MAP:LOCalization:STARt	258
MAP:LOCalization:STATe?	258
MAP:LOCalization:STOP	258
MAP:LOCalization:USE:CASE:PRESet	258
MAP:LOCalization:USE:CASE:PRESet?	259
MAP:MODE	259
MAP:MODE?	259
Narda Command Reference - HORizontal	261
HORizontal:SCAN:ADD:VALue	261
HORizontal:SCAN:CALC:RESult	261
HORizontal:SCAN:DATA?	261
HORizontal:SCAN:DELeTe:VALue	261
HORizontal:SCAN:DETeCtor	262

# Command Reference Guide

## Contents

---

HORizontal:SCAN:DETector? .....	262
HORizontal:SCAN:MANual:CORRection .....	262
HORizontal:SCAN:RESet .....	262
HORizontal:SCAN:RESet:MAX .....	262
HORizontal:SCAN:STARt .....	263
HORizontal:SCAN:STATe? .....	263
HORizontal:SCAN:STOP .....	263
HORizontal:SCAN:TYPE .....	263
HORizontal:SCAN:TYPE? .....	264
Narda Command Reference - PERSistence .....	265
PERSistence:DATA:ALL? .....	265
PERSistence:DATA:COLumn:COUNt? .....	265
PERSistence:DATA:FREQuency:STARt? .....	266
PERSistence:DATA:FREQuency:STEP? .....	266
PERSistence:DATA:OVERdriven? .....	266
PERSistence:DATA:REALtime? .....	267
PERSistence:DATA:ROW:COUNt? .....	267
PERSistence:DATA:UPDate? .....	267
PERSistence:MEASurement:TIME .....	268
PERSistence:MEASurement:TIME? .....	268
PERSistence:SCAN:COUNt .....	268
PERSistence:SCAN:COUNt? .....	269
PERSistence:SCAN:NUMBer? .....	269
PERSistence:TYPE .....	269
PERSistence:TYPE? .....	270
Narda Command Reference - Deprecated .....	271
BEARing:PERSistence .....	271
BEARing:PERSistence? .....	271
LEVelmeter:DETector:PERSistence .....	271
LEVelmeter:DETector:PERSistence? .....	271
LEVelmeter:TVIDeo .....	272
LEVelmeter:TVIDeo? .....	272
[SENSe:]APPLication:QUIT .....	272
[SENSe:]ATTenuator:ENTRy:MODE .....	272
[SENSe:]ATTenuator:ENTRy:MODE? .....	273
[SENSe:]REFerence:LEVel:ENTRy:MODE .....	273

---

[SENSe:]REFeRence:LEVel:ENTRy:MODE? .....	273
[SENSe:]RUN:SINgLe:OVERlap .....	274
[SENSe:]STOP:MODE .....	274
[SENSe:]STOP:MODE? .....	274
11 Appendix: Examples .....	277
SCPI Example – RT Spectrum .....	277
SCPI Example – Persistence .....	278
SCPI Example – Automatic DF .....	280
SCPI Example – VITA49 IQ Streaming .....	281
SCPI Example – VITA49 Spectrum Streaming via UDP Single Cast .....	282
SCPI Example – Audio Streaming via TCP .....	283
SCPI Example – Audio Streaming via UDP Single Cast.....	284

## 2 General Information

Following instructions are supplied together with the device:

- Quick Start Guide: These instructions offer a quick introduction on to how to use the SignalShark. They don't replace the detailed operation manual (Online help).
- Online help: The complete operation manual can be found on the device.
- General Safety Notes: Enclosed document General Safety Notes contains important information on how to avoid injury from incorrect use and on the correct handling of the product.

⇒ Be sure to read the Quick Start Guide and the General Safety Notes before operating the device and follow all instructions provided there.

⇒ Store all documents with the device and make them available to all users.

⇒ When transferring the device to third parties, also forward these instructions to them.

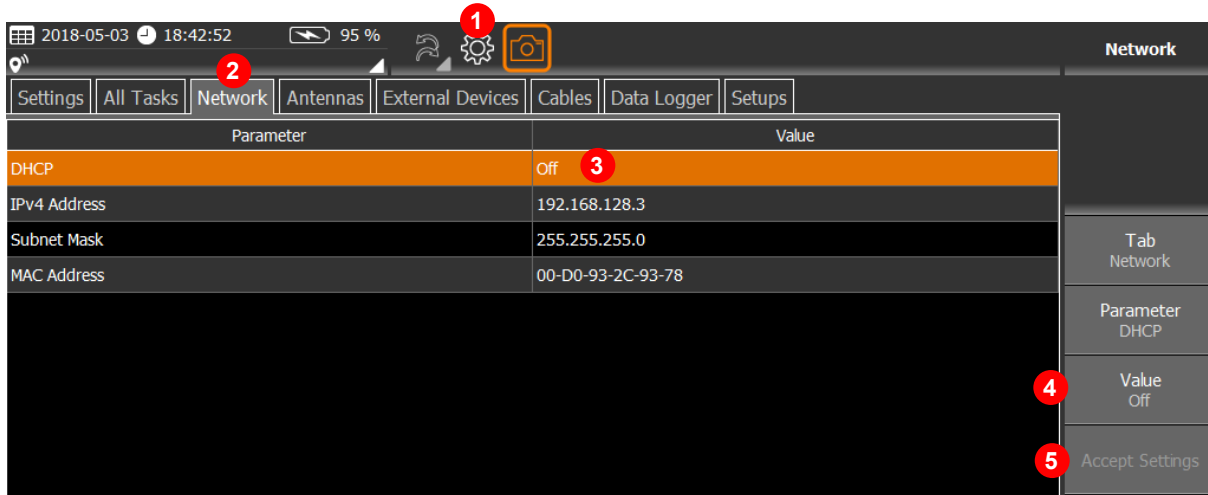
## SignalShark Network Setup

### Variable Network Settings

**IP-Address:** Ex-works set to DHCP

If you have administrative rights, you can change the IP address of the instrument as follows:

1. Open the "General Settings" menu
2. Select the "Network" tab
3. Select the desired parameter
4. Tap the "Value" button to change the value.
5. Tap on "Accept Settings" button to confirm changes
  - a. You have to enter the administrator password for the changes to take effect



## Fixed Network Setting

**Remote Control Port:** 5300  
**Termination character:** Carriage Return + Line Feed (CR+LF)

## Connecting to a network

The device can be remotely controlled using the Ethernet interface located on the top side panel of the device.



Setting up a connection:

1. Connect the SignalShark to the network using a commercially available network cable.
2. Adjust the network settings to match the network properties.
3. Open the network connection.



*You can check the connection state by sending the command **"\*IDN?"**. If connection was successful, the instrument will send basic information such as manufacturer, model, serial number and version.*

## Troubleshooting checklist

If connecting to the instrument fails:

- Check the connection (cables)
- Make sure that the SignalShark is switched on and provided with sufficient power
  - Batteries fully charged or powered via the AC adapter
- Check the settings of the IP address and the port
- Make sure there is no firewall preventing the communication
- Make sure the corresponding options are enabled

## 3 Narda SCPI basic principles

### Narda SCPI information

Some important basic information about how Narda uses SCPI is given below.

General SCPI basic principles can be found at: [www.ivifoundation.org/docs/scpi-99.pdf](http://www.ivifoundation.org/docs/scpi-99.pdf)

### Various parsers

Parser	Meaning
488	Parser for historic GPIB commands (e.g. *IDN?, *RST, *WAI).
SCPI99	Parser for universal SCPI commands (e.g. SYST:ERR?, ABORT).
NARDA	Narda-specific commands for measurements and measurement configuration. The Narda parser commands are derived from the GUI.

### Syntax convention

Symbol	Meaning
*	Always preceding a common command
:	Separates Keywords in different subsystems
;	Separates Keywords within the same subsystem
space	Separates Keyword from parameter
	Separates alternative commands (the bar is not sent with the command)
<>	Placeholder for numeric values (the angle brackets are not sent with the command)
[]	Placeholder for optional keywords/parameters (the square brackets are not sent with the command)
{ }	Placeholder for parameters that may be included 0 or more times (the curly braces are not sent with the command)
Keyword "Long form"	First letter in each word plus all letters in the last word. If only one word -> the whole word. The uppercase letters indicates the part of the keyword that is required which is also the short form of the keyword (see below).
Keyword "Short form"	Uppercase letters in the Long form indicates the corresponding Short form: <ul style="list-style-type: none"> <li>Keywords with 4 letters or less -&gt; all letters are used to form the short form</li> <li>Keywords with 5 or more letters where the 4<sup>th</sup> letter is <i>not</i> a vowel, the first 4 letters are used and if the 4<sup>th</sup> letter <i>is</i> a vowel the first 3 letter are used to form the short form.</li> </ul>



## Basic sequence for a remote-controlled measurement

With few exceptions, the procedure is as follows:

- Disable the GUI for high performance (SYSTem:REMOte:DISPlay OFF)
- Set the device to a defined state (e.g. \*RST or TASK:NEW)
- Configure the measurement task (create task, set parameters, etc.)
- RUN or HOLD the measurement (RUN:SINGLE, RUN:CONT, RUN:CONT:RES, HOLD)
- Update the measurement data (e.g. SPECTrum:DATA:UPDate?)
- Optional: HOLD or STOP the measurement
- Fetch the measurement data (e.g. SPECTrum:DATA:ALL?)
- Fetch the additional data (e.g. MARKer:SPECTrum:DATA:ALL?)

## Defined start state

The device must be set to a defined start state before remote operation (e.g. \*RST, SYST:ERR:CLEAR). It is a good idea to create a new task for each measurement task. Every new task is started with the default parameters (TASK:STATe? = RUN).

**When setting up the measurement parameters, the measurement should run (RUN) so that automatic parameter corrections are applied in the correct order.**

The task should be deleted after the measurement task is completed or a \*RST or TASK:NEW is performed before the next measurement task.

## End character

A maximum of 1460 bytes of user data can be transmitted in one go via TCP. Some of our data sets are very much bigger than this. It is therefore necessary to use end characters. Data without a valid end character will not be evaluated by the device. The end characters do not contain any user information and should be filtered out by the receiving program.

The Narda device sends <CR><LF> after each data packet.

The Narda device will accept data packets ending in <LF> or <CR> <LF>.

## Commands

Some commands are with parameters (SPEC:FREQ:CENt 100MHz) and some are without parameters (e.g. SENSE:STOP). Commands do not have a response. The control program does not therefore know when the command has been processed by the device. An error query (e.g. SYST:ERR?) or a request to read back the parameters (e.g. SPEC:FREQ:CENt?) should therefore be sent after each command or sequence of commands.

## Queries

SCPI queries always result in a response in Narda devices. Some queries are for reading out measurement parameters (e.g. SPEC:FREQ:CENt?) and some are for reading out the measurement data (e.g. SPEC:DATA:ALL?). There is no guarantee that the requested measurement data will be available at any given time, e.g., if a measurement has not been completed yet, or no data is available because of the measurement settings. A query delivers as the response either the requested parameters, or the measurement data, or NAN (not a number). NAN is defined in SCPI and is transmitted if no measurement data exist or the command is invalid at the time.

## Parameter corrections for Narda devices

It may be necessary for the device to correct certain parameters, for example, if an upper or lower limit is exceeded (e.g. Fcent 999GHz). Some parameters also affect each other (e.g. FSpan can affect the RBW and the measurement time). To avoid parameter corrections, the parameters must have valid values and be used in permitted combinations with each other. The affected parameters must be read back to test whether they have been corrected.

### Example

```
// Set parameters
SPEctrum:FREQuency:CENTer 10 MHz
SPEctrum:FREQuency:SPAN 1 GHz

// Query error
SYST:ERR:ALL?
0,"No error"

// Read back parameters (SPAN was corrected)
SPEctrum:FREQuency:CENTer?
10000000

SPEctrum:FREQuency:SPAN?
40000000
```

### SCPI error queue

The error queue contains the errors that have occurred in the command parser or when commands are processed. There may be several entries in the queue for each command and query. These entries can be queried individually (SYST:ERR?) or all together (SYST:ERR:ALL?). Depending on the measurement task, it may be useful to query the queue after every command or sequence of commands.

### SCPI and Narda error codes

- 400 to -499 SCPI query errors
- 300 to -399 SCPI device-specific errors
- 200 to -299 SCPI execution errors
- 100 to -199 SCPI command parser errors
- 0 No error
- > 0 Narda device-specific

### Standard error code format

Error code, "Error text"

### Extended Error Code Format

Error code, "Error text :Command and Parameter Information"

### Note concerning “:Command and Parameter Information“

The texts provide additional information to assist with troubleshooting. The content is not standardized and can change with a new firmware version. For this reason, the text should not be used for sequence control in control programs. Only the error code is provided for this purpose.

### Example

```
0,"No error"
110,"eERR_MEASUREMENT_TASK"
-224,"Illegal parameter value: 1234.569"
-113,"Undefined header: XYZ"
400,"eERR_FPGA: illegal BAND value"
```

## Task States (RUN, STOP, HOLD)



**To configure the measurement system, the measurement should run (RUN) so that automatic parameter corrections are applied in the correct order.** Every new task created will initially run using the default parameters in RUN mode. The measuring machine runs in RUN mode and stops automatically if the measurement is started with RUN:SINGLE. If the measurement is started with RUN:CONTinuous it can be stopped by SENSE:STOP or SENSE:HOLD. The measurement results should only be evaluated when the measuring machine is in STOP mode or HOLD mode. Measurement values continue to be recorded in the background in HOLD mode. These data are available for later evaluation.

## Measurement parameters and evaluation parameters

Some parameters affect the measurement (e.g. SPECTrum:FREQuency:CENTer, SPECTrum:FREQuency:SPAN) and other parameters affect data evaluation (e.g. MARKer:SPECTrum:FREQuency, MARKer:SPECTrum:SEARch:PEAK:EXCURsion).

The evaluation parameters can be used in HOLD and in STOP mode to perform a new calculation using the measurement data. If measurement parameters are changed, a new measurement must be performed in order to update the measurement data. If measurement parameters are changed in HOLD mode, the device switches to STOP mode automatically.

## Synchronizing several commands

For configuring the device, it is a good idea to subdivide the commands into groups (e.g. general parameters, spectrum parameters, peak table parameters). Several consecutive commands can be synchronized if a query follows each command or if the error queue is evaluated after a group of commands (e.g. SYST:ERR:ALL?).

### Example

// Synchronization of a command group by SYST:ERR:ALL?

```
FREQuency:CENTer 110 MHz
FREQuency:SPAN 40 MHz
RBW 100 kHz
SYST:ERR:ALL?
```

0,"No error"

// Synchronization of individual commands by reading back the parameters (Query)

```
FREQuency:CENTer 110 MHz
FREQuency:CENTer?
```

110000000

```
FREQuency:SPAN 40 MHz
FREQuency:SPAN?
```

40000000

```
RBW 100 kHz
RBW?
```

100000

### Concatenating commands and queries using ;

It is possible to send several commands in one line, each separated by a semicolon (;). The parser processes the commands from left to right, but it is possible that some combinations will produce unwanted interactions. The responses to queries are sent one at a time and must be assigned accordingly and evaluated by the receiver. At the end of such a line of commands, the error queue will contain all the errors from the individual commands and queries. It is therefore sensible to use this function only for special cases.

#### Example

// Set and read back a parameter

SPECtrum:RBW 100kHz; SPECtrum:RBW?

// Set several parameters and error query

FREQuency:SPAN 40 MHz;FREQuency:CENTer 110 MHz;SYSTem:ERRor:ALL?

### Remote Logfile

It is possible to save all remote data traffic in a log file on the device. This is an important troubleshooting tool. The log function can be activated or deactivated with the command SYSTem:REMote:LOG. Each activation overwrites the existing logfile. After each restart of the application, the logging function must be reactivated. The logfile is saved in the SignalShark log directory under D:\Narda\_Signalshark\Logfiles\Remotesystem.log.

#### Example Logfile:

```
2019-01-11 10:23:48.730 192.168.128.1:2852 +++
2019-01-11 10:23:53.337 192.168.128.1:2852 >>> *IDN?
2019-01-11 10:23:53.338 192.168.128.1:2852 <<< Narda Safety Test Solutions GmbH,SignalShark 3310,A-0054,V1.3.1
2019-01-11 10:23:53.356 192.168.128.1:2852 >>> *RST
2019-01-11 10:23:55.872 192.168.128.1:2852 >>> SYST:ERR:ALL?
2019-01-11 10:23:55.873 192.168.128.1:2852 <<< 0,"No error"
2019-01-11 10:26:20.340 192.168.128.1:2852 ---
```

### RUN:CONTInuous Commands

Continuous Run starts a continuous measurement. With each measurement run, the scan number is increased by one. The commands RUN:CONT? or RUN:CONT:RES? return a 0 if the start of the measurement was successful and return an error code if the start of the measurement was unsuccessful. RUN:CONT or RUN:CONT:RES writes an error in the error queue (SYSTem:ERRor?) if the start of the measurement was unsuccessful. A continuous measurement can be stopped with SENSE:STOP or SENSE:HOLD to evaluate measured data for example.

### RUN:SINGLe Commands

Single Run starts a measurement run until the scan number has reached the desired scancount. After that, the measurement automatically goes into STOP mode. The command RUN:SINGLe. starts the single run without waiting until the measurement is finished. The command RUN:SINGLe? waits until the measurement is finished. The command RUN:SINGLe? returns 0 if the measurement was successful and returns an error code if the measurement was unsuccessful. RUN:SINGLe writes an error in the error queue (SYSTem:ERRor?) if the measurement was unsuccessful.

### DATA:ALL? commands

The Data:ALL? commands consist of a header and optional data fields.

The header is always the same:

TimeStampSyncFlag, TimeStampSeconds, TimeStampFractional, ScanNumber

All other elements are optional and tagged with an ID. As soon as an ID is available, the associated data is also available. The ID is always followed by the number of associated data elements and then the data itself. Unknown IDs or IDs which are not evaluated and the associated data elements must be ignored during parsing. This allows a later extension with new IDs.

## Examples:

//Spectrum without IDs (ScanNumber=0)

SPECTrum:DATA:ALL?

0,0,0,0

//Spectrum with IDs: CONFIG, RMS, PPK

SPECTrum:DATA:ALL?

0,1532501199,579669619,4,CONFIG,5,1,1,101,31200000,400000,RMS,103,0,0,-90.36,-90.41,-91.52,-92.09,-91.00,-91.47,-92.21,-91.29,-90.17,-89.27,-87.97,-86.38,-86.33,-87.86,-88.66,-89.03,-88.81,-88.02,-87.16,-86.30,-85.64,-85.29,-84.98,-84.50,-83.84,-83.04,-82.09,-81.08,-80.18,-79.53,-79.28,-79.58,-80.38,-81.48,-82.60,-83.52,-84.31,-85.02,-85.61,-86.08,-86.48,-86.42,-86.15,-86.28,-86.23,-85.64,-85.12,-85.03,-84.89,-84.64,-84.53,-84.59,-84.74,-85.05,-85.59,-86.29,-86.89,-87.40,-88.16,-88.58,-88.40,-87.95,-87.56,-87.55,-87.82,-88.08,-88.25,-88.45,-88.50,-88.37,-88.28,-88.15,-87.87,-87.53,-87.05,-86.33,-85.53,-84.77,-84.19,-84.11,-84.42,-84.91,-85.52,-85.81,-85.15,-85.62,-87.70,-89.00,-89.57,-90.09,-90.68,-90.99,-91.23,-91.66,-91.73,-91.46,-91.12,-90.49,-89.74,-89.06,-88.49,PPk,103,0,0,-66.63,-66.77,-67.34,-68.57,-69.57,-67.28,-63.25,-59.96,-57.75,-56.64,-56.25,-56.35,-56.94,-57.77,-58.67,-59.61,-60.62,-61.87,-63.43,-65.24,-63.93,-63.50,-63.42,-63.56,-64.77,-64.82,-64.30,-63.25,-62.32,-61.49,-61.18,-61.35,-60.76,-59.63,-58.97,-58.54,-58.01,-57.67,-57.62,-57.22,-56.14,-54.92,-53.99,-53.43,-53.33,-53.74,-54.72,-56.16,-57.47,-58.35,-59.45,-61.58,-65.37,-67.96,-68.12,-68.47,-67.43,-65.38,-64.52,-64.31,-64.39,-64.27,-63.52,-61.91,-60.01,-58.52,-57.66,-57.35,-57.39,-57.64,-58.13,-58.88,-59.99,-61.62,-63.60,-65.90,-68.54,-68.58,-67.71,-66.97,-67.09,-67.69,-68.06,-68.50,-68.38,-69.05,-69.95,-65.79,-63.23,-61.94,-61.38,-61.06,-60.35,-58.69,-56.42,-54.16,-52.16,-50.52,-49.25,-48.25,-47.47

//Levelmeter without IDs (ScanNumber=0)

LEVelmeter:DATA:ALL?

0,0,0,0

//Levelmeter with IDs: PPK, RMS (without antenna handle)

LEVelmeter:DATA:ALL?

0,1532501012,938716127,195,PPk,4,0,0,-71.55,-71.55,RMS,4,0,0,-72.35,-72.35

//Levelmeter with IDs: COMPASS, PPK, RMS (with antenna handle)

LEVelmeter:DATA:ALL?

0,1532500912,935929584,430,COMPASS,3,275.7,-1.7,-94.1,PPk,4,0,0,-72.35,-72.35,RMS,4,0,0,-72.35,-72.35

//Peaktable without IDs (ScanNumber=0)

PEAKtable:DATA:ALL?

0,0,0,0

//Peaktable without IDs (no Peaks)

PEAKtable:DATA:ALL?

0,1532500432,350577299,5

//Peaktable with IDs: FREQUENCY, RMS, PPK (1 Peak)

PEAKtable:DATA:ALL?

0,1532500462,350577299,11,FREQUENCY,1,32859262.9335,RMS,3,0,0,-95.40,PPk,3,0,0,-88.28

## DATA:UPDate Commands



*It is recommended to use the HOLD mode together with the DATA:UPDate Commands, because then the readout memory is hold and so all measurement results can be queried synchronized one after the other. With each successful call of the DATA:UPDate Command, the readout memory is updated with the latest measurement results.*

Every new task created will initially run using the default parameters in RUN mode. The measuring machine runs in RUN mode and stops automatically if the measurement is started with RUN:SINGLE. If the measurement is started with RUN:CONTinuous it can be stopped by SENSE:STOP or SENSE:HOLD. The measurement results should only be evaluated when the measuring machine is in STOP or HOLD mode. Measurement values continue to be recorded in the background in HOLD mode. These data are available for later evaluation.

The DATA:UPDate commands are used to update the measurement data automatically. This is possible in HOLD mode and in RUN mode.

The update mechanism has 2 modes controlled by the scan number parameter:

Mode 1 Always Forcing an new update:

Scan number parameter must be empty or NAN

Mode 2 Update only if actual scan number is greater then last scan number:

Scan number parameter must be the last scan number of the measurement application

For the first update the scan number parameter must be 0 or empty

For very long measurements, a timeout can be specified.

### Example Mode 1:

Command Timeout = Default (3s)

Spectrum Measurement Time = 1s

SENSe:HOLD

SPEC:DATA:UPD?

22

SPEC:DATA:LEVel? RMS

-92.86,-78.46,-60.11,-50.56,-47.57,-50.56,-60.11,-78.32,-86.26

SPEC:DATA:UPD?

23

SPEC:DATA:LEVel? RMS

-92.84,-78.43,-60.09,-50.55,-47.56,-50.55,-60.10,-78.29,-86.26

SPEC:DATA:UPD?

24

SPEC:DATA:LEVel? RMS

-92.88,-78.45,-60.09,-50.55,-47.55,-50.55,-60.10,-78.29,-86.01

### Example Mode 2:

Command Timeout = 200ms  
Spectrum Measurement Time = 1s

SENSe:HOLD

SPEC:DATA:UPD? 0,0.2

22

SPEC:DATA:LEVel? RMS

-92.49,-78.13,-59.89,-50.38,-47.39,-50.38,-59.91,-78.10,-86.80

SPEC:DATA:UPD? 22,0.2

22

SPEC:DATA:UPD? 22,0.2

22

SPEC:DATA:UPD? 22,0.2

22

SPEC:DATA:UPD? 22,0.2

22

SPEC:DATA:UPD? 22,0.2

23

SPEC:DATA:LEVel? RMS

-92.52,-78.27,-59.98,-50.45,-47.45,-50.45,-59.99,-78.25,-87.20

## Commands with long Timeouts

Some commands have long timeouts as specified in the documentation. For these commands, the maximum command runtime is longer than for other commands. If a timeout = 10s is given in the documentation, this means that the maximum processing time in the device is 10s. If this time is exceeded, there is a processing error. In order to avoid blocking of the following commands due to the long processing times of a command, a query should be executed after such a command (for example, SYST:ERR:ALL?). The default timeout is 3 seconds for commands without special specification in the documentation and can be queried using SYSTem:REMOte:TIMEout?.

## Progress query for long Timeouts

A progress query and adjusting the timeout only makes sense in special cases.

For some settings (for example, SETTings:FAST:SCANs:ONLY false), long initialization times (measurement parameter calculation) and long measurement times may occur. It is advisable to stop the measurement before parameter changes. In that way a long timeout only occur once when starting the measurement and not with every parameter change.

## Command Reference Guide

Narda SCPI basic principles

---

### Example:

Parameter change in Scan Spectrum with a huge amount of bins (Slow Scan).  
Using progress query command to query current calculation progress.

```
// Stop Measurement
SENSe:STOP
SYSTem:ERRor:ALL?
0,"No error"

// Change parameters of Scan Spectrum measurement
SPEctrum:FREQuency:STARt 8 kHz
SPEctrum:FREQuency:STOP 8 GHz
SPEctrum:RBW 1 kHz
SYSTem:ERRor:ALL?
0,"No error"

// Start Single Measurement with timeout of 3 seconds
// The measurement parameter calculation begins before the measurement starts
// In this example parameter calculation takes longer than 3 seconds, so we got an timeout error
SENSe:RUN:SINGle 3s
SYSTem:ERRor:ALL?
-365,"eERR_TIME_OUT"

// Optional: Cyclic query of the command progress
// When measurement parameter calculation is finished,the measurement starts and the response of the command
// progress query is 0.0, "".
SYST:COMM:PROG?
320,1250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
743,1250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
1065,1250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
1207,1250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
0,0,""

// Wait until the first measurement is completed and query the measurement results
SPEC:DATA:UPD? 0
NaN

SPEC:DATA:UPD? 0
NaN

SPEC:DATA:UPD? 0
1

SPEC:DATA:LEV? RMS
-94.26,-79.34,-61.02,-51.51,-48.54,-51.53,-61.07,-79.24,-86.54
```



```
// Stop Measurement
SENSe:STOP
SYSTem:ERRor:ALL?
0,"No error"

// Change parameters of Scan Spectrum measurement
SPECTrum:RBW 10 kHz
SYSTem:ERRor:ALL?
0,"No error"

// Start Continuous Measurement with timeout of 0 seconds
// The measurement parameter calculation begins before the measurement starts
SENSe:RUN:CONTinuous 0s
SYSTem:ERRor:ALL?
0,"No error"

// Optional: Cyclic query of the command progress
//When measurement parameter calculation is finished, the measurement starts and the response of the command
//progress query is 0.0, ""
SYST:COMM:PROG?
151,250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
228,250,"SPECTRUM_CALC_SCAN_STEPS"

SYST:COMM:PROG?
0,0,""

// Wait until the first measurement is completed and query the measurement results
SPEC:DATA:UPD? 0
NaN

SPEC:DATA:UPD? 0
NaN

SPEC:DATA:UPD? 0
1

SPEC:DATA:LEV? RMS
-94.26,-79.34,-61.02,-51.51,-48.54,-51.53,-61.07,-79.24,-86.54
```

## GUI concept as template for the Narda parser

The commands and queries for the Narda parser are closely linked to the device GUI. As a result, sequences can be described using the device GUI and implemented for remote control relatively easily. The commands are structured as defined in the GUI. Commands in the uppermost operating level are prefixed by [SENSe:]. Commands that are not defined in the GUI are prefixed by SYSTem: Some parameters are valid for all tasks (e.g. [SENSe:]ATTenuator) and some are only valid for specific tasks (e.g. SPECTrum:SCAN:COUNT). There are also some general settings (e.g. SETTings:DATE?) and information (e.g. DEVIce:APPLication:VERSion?).

# 4 Narda SCPI Data types

The following data types are supported by the NARDA SCPI Parser:

## Arbitrary

Any combination of all data types

## Binary

Format: #<Headersize>[Binarysize][Binarydata]

Example: #213binarydata123

## Bool

Input: 0,1,OFF,ON,FALSE,TRUE,NO,YES

Output: 0,1

## ByteArray

Hexadecimal coded strings with the following formats are supported:

**Example Format 0:** "AABBCCDD00224455" (Hexdump)

**Example Format 1:** "AA,BB,CC,DD,00,22,44,55" (8Bit)

**Example Format 2:** "AABB,CCDD,0022,4455" (16Bit)

**Example Format 4:** "AABBCCDD,00224455" (32Bit)

The following special cases are supported:

"1,2,A,B" results in "01,02,0A,0B"

When the input format and the output format differ:

"DD00224455" (Format 0) results in "DD,00,22,44,55" (Format 1)

"DD00224455" (Format 0) results in "00DD,0022,4455" (Format 2)

"DD00224455" (Format 0) results in "000000DD,00224455" (Format 4)

When the input format and the output format are reversed (Byte swapping):

"DD00224455" (Format 0) results in "DD00,2200,5544" (Format 2)

"DD00224455" (Format 0) results in "DD000000,55442200" (Format 4)

The following formats are NOT supported:

"0x1234", "0X1234", "x1234", "00,ABCD,EE,00112233", "00 11 22 33"

## Char

Range: -128...127

## Date

Input Format [DAY\\_MONTH\\_YEAR](#): "dd.MM.yyyy"

Input Format [MONTH\\_DAY\\_YEAR](#): "MM/dd/yyyy"

Input Format [YEAR\\_MONTH\\_DAY](#): "yyyy.MM.dd"

Input Format [YEAR\\_MONTH\\_DAY\\_ISO](#): "yyyy-MM-dd"

Output Format (Default) = [YEAR\\_MONTH\\_DAY\\_ISO](#)

## Datelso

Input Format `YEAR_MONTH_DAY_ISO`: "yyyy-MM-dd"

Output Format (Default) = `YEAR_MONTH_DAY_ISO`

## Double

Input String Format:

e format as[-]9.9e[+] -]999

E format as[-]9.9E[+] -]999

f format as[-]9.9

g use e or f format, whichever is the most concise

G use E or f format, whichever is the most concise

Input Precision:

A precision is also specified with the argument format. For the 'e', 'E', and 'f' formats, the precision represents the number of digits after the decimal point. For the 'g' and 'G' formats, the precision represents the maximum number of significant digits (trailing zeroes are omitted).

Default Output Format: 'g'

Default Output Precision: 12

## Enum

**Examples:** STOP, HOLD, RUN, LEFT, RIGHT, RMS, AVERAGE

## FixArray

FixArrays can be composed of different data types. The first parameter specifies the number of elements.

**Example FixArray[Double]:** 3,22.09765625,33.09765625,44.09765625

**Example FixArray[UShort]:** 3,110,120,130

**Example FixArray[Date,String]:** 1,23.01.1981,"Hello World"

## FlexArray

Flexarrays can be composed of different data types.

Flexarrays can only occur at the end of a parameterlist.

**Example FlexArray[Double]:** 22.09765625,33.09765625,44.09765625

**Example FlexArray[UShort]:** 110,120,130

**Example FlexArray[Date,String]:** 23.01.1981,"Hello World"

## Optional

Optional parameters can be composed of different data types.

Optional parameters can only occur at the end of a parameter list.

Optional parameters can also be empty.

**Example Optional[Enum]:** RMS,PPk,MPk or empty

**Example Optional[Timespan]:** 100ms or empty

## Float

See Double

Default Output Format: 'g'

Default Output Precision: 7

### Frequency

See Double

Input Format: Hz, kHz, MHz, GHz, THz, no unit (=Hz)

Output Format: no unit (=Hz)

**Example:** 100e3, 100kHz, 100 kHz, 100000

### FrequencyEnum

See Frequency

Input Format: Hz, kHz, MHz, GHz, THz, no unit (=Hz)

Input Format Enum: MIN, MINIMUM, MAX, MAXIMUM, UP, DOWN

Output Format: no unit (=Hz)

**Example:** 100e3, 100kHz, 100 kHz, 100000, MIN, MAX

### LatLon

Input Format DEGREE

**Examples:** -27.46758,153.02789

Input Format DEGREE\_WITH\_HEMISPHERE

**Examples:** 27.46758S,153.02789E

Input Format DEGREE\_MINUTES

**Examples:** -27D28.055',153D1.673'

Input Format DEGREE\_MINUTES\_WITH\_HEMISPHERE

**Examples:** 27D28.055'S,153D1.673'E

Input Format DEGREE\_MINUTES\_SECONDS

**Examples:** -27D28'3.3",153D1'40.4"

Input Format DEGREE\_MINUTES\_SECONDS\_WITH\_HEMISPHERE

**Examples:** 27D28'3.3"S,153D1'40.4"E

Output Format (Default) = DEGREE

### Long

Range: -2147483648 ... 2147483647

### Short

Range: -32768 ... 32767

### String

**Example:** "Hello World"

**Example:** 'Hello World'

**Example:** "Hello 'b,i,g' World"

**Example:** 'Hello "b,i,g" World'

## String488

String including only the following characters:

Letters: [A-Z] [a-z]

Numbers: [0-9]

Special characters: [ -/.]

**Example:** "Hello-W.o.r.l.d/ 1 2 3"

## Time

Input Format 24H: "hh:mm:ss"

Input Format 12H: " hh:mm:ss AM | PM"

Output Format (Default) = 24H

## Timespan

See Double

Input Format: ms, h, m, s, no unit (=s)

Output Format: no unit (=s)

**Example:** 10e3 ms, 100s

## UChar

Range: 0 ... 255

## ULong

Range 0 ... 4294967295

## ULongEnum

See ULong

See Enum

Input Format: Enum or ULong

## ULongLong

Range 0 ... 18446744073709551615

## Unit

See Double

Input with no unit = Currently set unit

Input World A: dBm, dBV, dBmV, dBuV

Input World B: W\_m2, W\_cm2, A\_m, V\_m, dBA\_m, dBV\_m, dBmV\_m, dBuV\_m

Input Word C: %

Input World D: A, dBA

Input Attenuation: dB

Output = Currently set unit

**Example:** 10dB, 10.3dBm

### UShort

Range 0 ... 65535

## 5 Stream Communication

### Definitions

The following terms are used to describe the streaming communication.

Term	Explanation
Stream	A Stream consists of one or more data packets transmitted automatically one after the other.
Stream Identifier (ID)	The Stream ID determines which data packets belong to a particular Stream. Several Streams transmitted in parallel can each be uniquely identified using the Stream ID in the Header.
Data packet	A data packet is transmitted as an entity and consists of Header, Context and Data items. The Context or Data items may be empty in some cases.
Header	The Header is a defined data structure at the start of each data packet and has the same format for all Streams that are being transmitted.
Context	The Context differs from Stream to Stream and contains additional information about the data items.
Data item	Data items are the actual payload data in a data packet and differ in structure and number from Stream to Stream. An individual data item consists of a single value or a structure containing various values, depending on the Stream.
Data item format	The data item format determines the basic data types (e.g. INT16, FLOAT32) that make up a data element.
Packet Counter	The Packet Counter numbers the individual data packets consecutively. This enables the receiver to determine whether all the data packets were received.
Endianness	Endianness describes the way that the bytes in the basic data types (e.g. INT16, FLOAT32) are arranged in memory. The byte arrangement for Microsoft Windows based systems is little endian.
Word	32 bit or 2 bytes
Streamlink	The Streamlink is a data connection for transmitting a Stream.
Remotelink	The Remotelink is used to configure and control the device settings and the Stream transmitted by the device.

### General information

A Stream consists of one or more data packets transmitted automatically one after the other. A data packet is transmitted as an entity and consists of Header, Context and Data items. The Context or Data items may be empty in some cases.

### Example: Structure of a data packet

Header	Context	Data items
--------	---------	------------

The Header is identical for all data packets in a Stream.

## Command Reference Guide

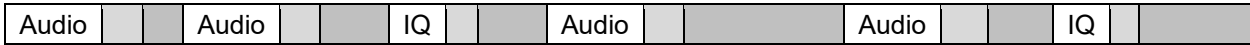
### Stream Communication

The Stream ID in the Header determines the size and structure of the Context and the Data items. The Header, Context and data items are always transmitted together as a packet.

#### Note

Unknown packets have to be filtered out and ignored using the Header information.

#### Example: IQ Data packets in a Stream



#### Streamlink and Remotelink

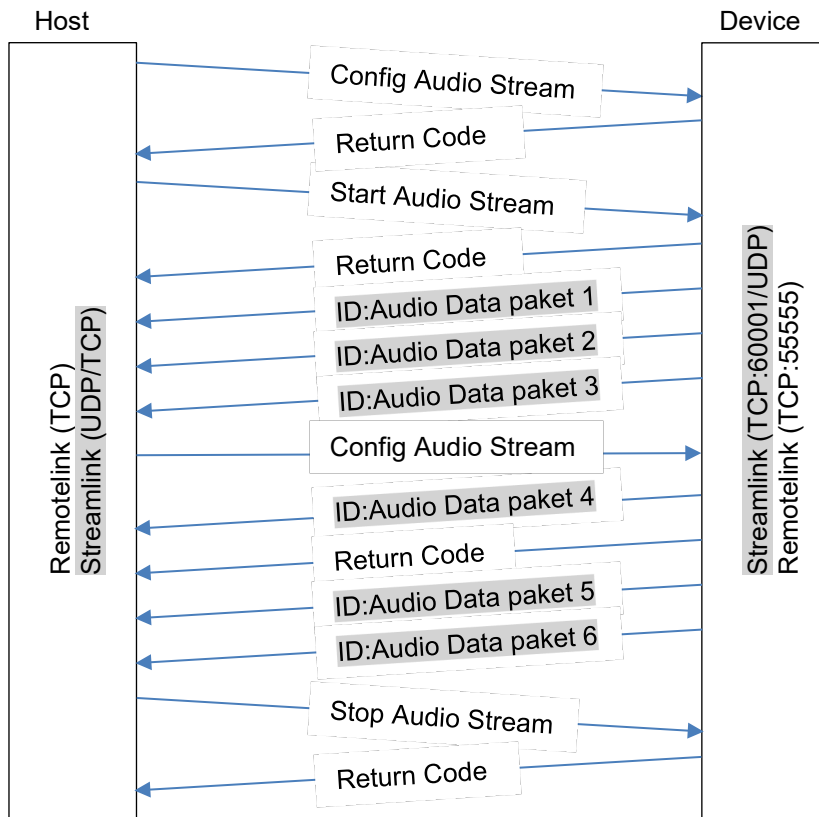
A Remotelink and a Streamlink are used for Stream communication.

The Remotelink consists of the ASCII remote commands described in this document.

The Streamlink consists of the binary data packets for one or more Streams.

The Streamlink is configured and controlled by the Remotelink.

#### Example: Remotelink and Streamlink between host and device





## General Stream Header

### Description

The Header is a defined data structure at the start of each individual data packet and has the same format for all the Streams that are being transmitted.

Together with the Stream Version, the Stream ID is used to uniquely identify the data contents.

This makes it possible to implement a receiver routine that distinguishes between different Stream IDs and Stream Versions so they can be processed.

The packet counter is used to verify the transmitted data. Each data packet transmitted is numbered consecutively before it is sent. This enables the receiver to detect whether data packets have been lost or are out of sequence.

### Header format

Parameter	Type	Bytes	Description
ByteOrder	Word	2	0x55AA = little endian 0xAA55 = big endian (not supported)
HeaderVersion	Word	2	Header version 0x0001 = Stream Header Version 1
StreamID	Word	2	Stream identifier 0x0002 = Stream Audio
StreamVersion	Word	2	Version for context and data items 0x0002 = Version 1 (of e.g. Stream Audio)
Reserved	Word	2	Reserved for later use
Reserved	Word	2	Reserved for later use
PacketCounter	DWord	4	Packet counter for lost packet detection
SizeOfContext	DWord	4	Number of bytes for the context data
NumberOfItems	DWord	4	Number of data items
SizeOfItem	DWord	4	Size of a single data item
Stream identifier	DWord	4	Set a run time by the user

## 6 Narda Audio Stream Context

### Description

The Audio Stream Context is a defined data structure which is transmitted in an Audio data packet after the Header. The size of the Context structure is specified by the SizeOfContext data field in the Header.

The timestamp consists of the two data fields IntegerSeconds and FractionalSeconds.

The time is based on the device RTC (real time clock). The RTC must be set anew after every device start using remote commands (DATE; TIME;). The timestamp is synchronized with the RTC once when the Audio Stream is started and is then computed with sample accuracy for each transmitted data packet until the Audio Stream is stopped.

The EventFlags 0x00000001, 0x00000002, 0x00000004 and 0x00000008 are important for the assessment of measurement result quality. The EventFlags bit 0x00000004 is set after every parameter change and remains set until the measurement data capture function is stable again.

### Audio stream context format

Parameter	Type	Bytes	Description
IntegerSeconds	DWord	4	Integer second part of the timestamp in UTC seconds since January 1, 1970 (without leap seconds).
FractionalSeconds	DWord	4	Nanosecond part of the timestamp to add to the second part.
Reserved	DWord	4	Reserved for later use
Reserved	DWord	4	Reserved for later use
DataltemFormat	Word	2	0x0002 = INT16 (16 Bit Audio)
DemodMode	Word	2	0x0001 = FM mono 0x0002 = AM 0x0003 = CW 0x0004 = LSB 0x0005 = USB 0x0006 = IQ 0x0007 = ISB 0x0008 = PULSE 0x0009 = AM and FM mono 0x0010 = PM
Channels	Word	2	Number of audio channels
BFO	Float	4	Beat Frequency [Hz]
Reserved	Float	4	Reserved for later use
Squelch Value	Float	4	Squelch Threshold Value
SampleRate	Float	4	Audio sample rate [Hz]
CBW	Float	4	Audio bandwidth [Hz]
Ftune	Double	8	Demodulation frequency [Hz]
Attenuator	Float	4	Attenuator 0...50 dB

## 7 Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

This chapter includes information about how Vita49.2 is used with Narda SignalShark. Detailed information about Vita 49.2 can be found in the ANSI/VITA 49.2-2017 Documentation.

### Spectrum Data Packet Class – Overview

Packet Header		
Parameter	Selected Option	Comments
Packet type	Signal Data Packet with stream ID	Conveys the digitized spectrum data
Packet Size	Variable	Depends on receiver settings.
Stream identifier	Yes	Set at run time by the user
Class ID	No	
Integer-seconds timestamp	UTC	Present in every packet
Fractional seconds timestamp	Real Time	

Packet Payload		
Parameter	Selected Option	Comments
Packing method	Processing efficient	
Item packing field size	16	Specified by data packet payload field in the paired Spectrum Context Packet.
Data item size	16	Specified by data packet payload field in the paired Spectrum Context Packet.
Event-tag size	0 bits	No event tags
Channel-tag size	0 bits	No channel tags
Vector size	0 bits	No vector
Real/complex type	Log Power Data in dB relative to reference level, real data	Spectrum bins
Data item format	Signed fixed point	[1,8,7] (7 fractional bits, 8 integral bits, 1 bit for sign)
Sample repeating/channel repeating	No repeating	No repeating of any kind
Repeat count	0	

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

Packet Trailer		
Parameter	Selected Option	Comments
Calibrated time indicator	Yes	0: timestamp is free running 1: timestamp is related to some external reference time
Valid data indicator	Yes	0: data invalid, 1: data valid Is set to 1 at detection of the “data invalid flag” OR “overflow flag” within the internal spectrum data
Reference lock indicator	No	
AGC/MGC indicator	No	
Detected signal indicator	No	
Spectral inversion indicator	No	
Over-range indicator	Yes	0: No Over-range, 1: Over-range Is set to 1 at detection of the “overdriven flag” within the internal spectrum data stream
Sample loss indicator	Yes	0: No Sample loss, 1: Sample loss Is set to 1 in the first packet after detection of a “frame number gap” > 1 within the internal spectrum stream
Sample Frame Indicators	Yes	Always 0 for RT Spectrum i.e. data is send within a single frame/packet
User-defined indicators	No	
Associated context packet count	Yes	0: No associated context packet 1: A context packet is send before every first data packet of a complete spectrum  Usually, one Spectrum context packet is sent during one data packet interval. Context packages are always transmitted before the associated data packet

## Spectrum Context Packet Class – Overview

Packet Header		
Parameter	Selected Option	Comments
Packet type	Spectrum Context	
Stream identifier	Yes	Identical to the stream identifier of the associated Spectrum data stream.
Class ID	No	
Integer-seconds timestamp	UTC	Present in every packet
Fractional seconds timestamp	Real Time	
Timestamp precision	Packet precision	Context changes apply to the sampling interval of the data packet with the exact same timestamp (TSM = 1)

Context Fields		
Parameter	Selected Option	Comments
Context field change indicator	Yes	0: no context field changes 1: at least one context field change with respect to last conveyed packet
Reference point identifier	No	
Bandwidth	No	
IF reference frequency	No	
RF reference frequency	Yes	Fcent for RT-Spectrum or Fstart of a Scan Spectrum
RF reference frequency offset	No	
IF band offset	No	
Reference level	Yes	Level in dBm of a CW signal at the frontend of the compensated signal chain which generates full scale spectrum data.
Gain	No	
Over-range count	No	
Sample rate	Yes	Sample Rate in Hz
Timestamp adjustment	No	
Timestamp calibration time	No	
Temperature	No	
Device identifier	No	
State and event indicators	No	Not supported. Pls. use indicator flags of the data packet instead
Data packet payload	Yes	
Logical events	No	
Calibrated time indicator	No	
Valid data indicator	No	
Reference lock indicator	No	
AGC/MGC indicator	No	
Detected signal indicator	No	
Spectral inversion indicator	No	
Over-range indicator	No	
Sample loss indicator	No	
User-defined indicators	No	
GPS	Yes	Is transmitted only when available
Formatted GPS geolocation	Yes	Altitude represents height above mean sea level
Formatted INS geolocation	No	
ECEF ephemeris	No	
Relative ephemeris	No	
GPS ASCII	No	

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

Ephemeris reference identifier	No	
Context association lists	No	
Source list	No	
System list	No	
Vector component list	No	
Asynchronous channel list	No	
Spectrum Type	Yes	R = 0, Window Time Interpretation = 2 (samples), Averaging Type = 1 (RMS), 2 (+Pk) or 4 (-Pk), Spectrum Type = 1 (log Power in dBr)
Window Type	Yes	100 (user defined by Narda : 4 term Nuttall with 3 degrees of freedom)
Npoints	Yes	FFT Size
Wnpoints	Yes	Window length in Samples
Resolution	Yes	Bin width in HZ
Span	Yes	FSpan in Hz
Naverages	Yes	Number of averaged spectra
Weighting Factor	No	Unused = 0
F1	Yes	Index of the first bin relative to the index of the bin at reference frequency
F2	Yes	Index of the last bin relative to the index of the bin at reference frequency
Window Time-Delta	Yes	time difference between subsequent FFTs in samples

## Spectrum Data Packet Class – Details

The packet is transmitted in big-endian order, most significant byte first.

Packet Field Name	Size in Words	
Header	1	
Stream Identifier	1	
Integer-seconds Timestamp	1	
Fractional-seconds Timestamp	2	
Data Payload	Depends on settings for spectrum streaming	
Trailer	1	

### Header (1 Word)

Bits	Value	Description
4	b0001	<b>Packet Type =</b> Spectrum Data packet with Stream Identifier

1	0	<b>Class ID bit (C bit)</b> Spectrum Data packets not include the Class ID field
1	1	<b>Trailer bit (T bit)</b> Spectrum Data packets include the trailer field
1	1	<b>Not Vita 49.0</b>
1	1	<b>Spectral Data</b>
2	b01	<b>Timestamp integer (TSI) =</b> Coordinated Universal Time (UTC)
2	b10	<b>Timestamp fractional (TSF) =</b> Real Time (Picoseconds) Timestamp
4	0...15	<b>Packet Count</b> Modulo-16 count of Spectrum Data packets.
16	big-endian	<b>Packet Size</b> This 16-bit field indicates the total number of 32-bit words present in the Spectrum Data packet, including the header, payload and all other fields.

### Stream Identifier (1 Word, big-endian)

The Stream Identifier (Stream ID) is a 32-bit number assigned to a VRT Packet Stream. The Stream Identifier can be set by user and is used to separate different streams from each other.

### Integer-seconds Timestamp (1 Word, big-endian)

The Integer-seconds Timestamp conveys UTC and provides the Reference-Point Time of the first Data Sample in the packet in seconds, including leap seconds, since midnight January 1, 1970, Greenwich Mean Time.

### Fractional-seconds Timestamp (2 Words, big-endian)

The Real-Time Timestamp extends the resolution of the Integer-seconds Timestamp down to one picosecond.

### Data Payload

The number of words in the data payload is variable and depends on the settings. The format of the data items 16 bit fixed point. Data items are send as log power data in dB relative to reference level.

### 16 Bit Payload Format (big-endian)

signed 16 bit fix point number

Bits	Description
1	Sign bit
8	Integral part
7	Fractional part

### Trailer (1 Word)

Bits	Value	Description
1	1	Enables Calibrated Time Indicator
1	1	Enables Valid Data Indicator

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

1	0	Enables Reference Lock Indicator (not implemented)
3	0	Reserved
1	1	Enables Over-range Indicator
1	1	Enables Sample Loss Indicator
4	0	Reserved
1	1	Calibrated Time Indicator
1	1 or 0	Valid Data Indicator
1	1 or 0	Reference Lock Indicator (not implemented)
3	0	Reserved
1	1 or 0	Over-range Indicator
1	1 or 0	Sample Loss Indicator
2	0	Sample Frame Indicator (0 for RT Spectrum),
1	1	Enable Associated Context Packet Count
7	1 or 0	Associated Context Packet Count

## Spectrum Context Packet Class - Details

The packet is transmitted in big-endian order, most significant byte first.

Packet Field Name	Size in Words
Header	1
Stream Identifier	1
Integer-seconds Timestamp	1
Fractional-seconds Timestamp	2
Context Indicator Field 0	1
Context Indicator Field 1	1
RF Reference Frequency	2
Reference Level	1
Sample Rate	2
Data Packet Payload Format	2
Formatted GPS (optional)	11
Spectrum Type	1
Window Type	1
Npoints	1
Wnpoints	1
Resolution	2
Span	2
Naverages	1
Weighting Factor	1
F1	1
F2	1
Window Time-Delta	1



## Header (1 Word)

Bits	Value	Description
4	b0100	<b>Packet Type</b> = Signal Data context packet
1	0	<b>Class ID bit (C bit)</b> Signal Data packets not include the Class ID field
1	0	<b>Reserved</b>
1	1	<b>Not a V49.0 Packet Indicator</b>
1	1	<b>Timestamp Mode (TSM)</b> Context changes apply to the sampling interval of the data packet
2	b01	<b>Timestamp Integer (TSI) =</b> Coordinated Universal Time (UTC)
2	b10	<b>Timestamp Fractional (TSF) =</b> Real Time (Picoseconds) Timestamp
4	0...15	<b>Packet Count</b> Modulo-16 count of Spectrum Data packets.
16	big-endian	<b>Packet Size</b> This 16-bit field indicates the total number of 32-bit words present in the Spectrum context packet, including the header and all other fields.

## Stream Identifier (1 Word, big-endian)

The Stream Identifier (Stream ID) is a 32-bit number assigned to a VRT Packet Stream. The Stream Identifier can be set by user and is used to separate different streams from each other.

## Integer-seconds Timestamp (1 Word, big-endian)

The Integer-seconds Timestamp conveys UTC and provides the Reference-Point Time of the first Data Sample in the packet in seconds, including leap seconds, since midnight January 1, 1970, Greenwich Mean Time.

## Fractional-seconds Timestamp (2 Words, big-endian)

The Real-Time Timestamp extends the resolution of the Integer-seconds Timestamp down to one picosecond.

## Context Indicator Field 0 (1 Word)

The Context Indicator field contains bit fields, one for each Context Field, that indicate whether the corresponding optional Context field is present in the packet.

Bits	Value	Description
1	0 or 1	Context Field Change Indicator
1	0	Reference Point Identifier
1	0	Bandwidth
1	0	IF Reference Frequency
1	1	RF Reference Frequency
1	0	RF Reference Frequency Offset

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

1	0	IF Band Offset
1	1	Reference Level
1	0	Gain
1	0	Over-range Count
1	1	Sample Rate
1	0	Timestamp Adjustment
1	0	Timestamp Calibration Time
1	0	Temperature
1	0	Device Identifier
1	0	State/Event Indicators
1	0	Signal Data Packet Payload Format
1	0 or 1	Formatted GPS
1	0	Formatted INS
1	0	ECEF Ephemeris
1	0	Relative Ephemeris
1	0	Ephemeris Ref ID
1	0	GPS ASCII
1	0	Context Association Lists
1	0	Field Attributes Enable
3	0	Reserved for CIF expansion
1	0	CIF 3 Enable
1	0	CIF 2 Enable
1	1	CIF 1 Enable
1	0	Reserved

### Context Indicator Field 1 (1 Word)

The Context Indicator field contains bit fields, one for each Context Field, that indicate whether the corresponding optional Context field is present in the packet.

Bits	Value	Description
1	0	Phase Offset
1	0	Polarization
1	0	3-D Pointing vector
1	0	3-D Pointing Vector Structure
1	0	Spatial Scan Type
1	0	Spatial Reference Type
1	0	Beam width
1	0	Range (Distance)
3	0	Reserved
1	0	$E_b/N_o$ BER
1	0	Threshold
1	0	Compression Point
1	0	2 <sup>nd</sup> and ThirdOrder Intercept Points

1	0	SNR/Noise Figure
1	0	Aux Frequency
1	0	Aux Gain
1	0	Aux Bandwidth
1	0	Reserved
1	0	Array of CIFS
1	1	Spectrum
1	0	Sector Scan/Step
1	0	Reserved
1	0	Index List
1	0	Discrete I/O (32 bit)
1	0	Discrete I/O (64 bit)
1	0	Health Status
1	0	V49 Spec Compliance
1	0	Version and Build Code
1	0	Buffer Size
1	0	Reserved

### RF Reference Frequency (2 Word, big-endian)

Is an unsigned 64bit fixpoint number.

Bits	Description
44	Integer Part
20	Fractional Part

### Reference Level (1 Word, big-endian)

32-bit format shown in the table below. The upper 16 bits of this field are reserved and should be set to zero. The Reference Level value is expressed in two's-complement format in the lower 16 bits of this field. This field has an integer and a fractional part, with the radix point to the right of bit 7.

Bits	Description
16	Reserved = 0
1	Sign bit
8	Integer Part
7	Fractional Part

### Sample Rate (2 Word, big-endian)

Is an unsigned 64bit fixpoint number.

Bits	Description
44	Integer Part
20	Fractional Part

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

### Data Packet Payload Format (2 Words)

Bits	Value	Description
1	0	Packing Method = processing-efficient packing
2	b01	Data Sample Type = Complex, Cartesian
5	b00000	Data Item Format = Signed Fixed-Point
1	0	Repeat Indicator = 0
3	0	Event-Tag Size = 0
4	0	Channel-Tag Size = 0
4	0	Reserved
6	b001111 or b011111	Item Packing Field Size 001111 = 16-1 for 16 Bit Format 011111 = 32-1 for 32 Bit Format
6	b001111 or b011111	Data Item Size 001111 = 16-1 for 16 Bit Format 011111 = 32-1 for 32 Bit Format
16	0	No Repeat Count
16	0	Vector Size = 0

### Formatted GPS (optional) (11 Words)

Bits	Value	Description
4	0	Reserved
2	b00	Timestamp Integer (TSI)
2	b00	Timestamp Fractional (TSF)
24	0	GPS/INS Manufacturer OUI
32	big-endian	Integer second timestamp of Position Fix
64	big-endian	Fractional second timestamp of Position Fix
32	big-endian	Latitude in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Longitude in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Altitude in meters, signed 32 bit fix point number, 27 bit Integer and 5 bit fractional part
32	big-endian	Speed over Ground in meters/second, unsigned 32 bit fix point number, 16 bit Integer and 16 bit fractional part
32	big-endian	Heading Angle in degrees, (not implemented, value = 0x7FFFFFFF)
32	big-endian	Track Angle in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Magnetic Variation, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part

### Spectrum Type (1 Word)

Bits	Description
12	Reserved
4	Window Time
8	Averaging Type
8	Spectrum Type

### Window Type (1 Word)

Bits	Description
16	Reserved
8	Reserved
8	Window Type

### Npoints (1 Word)

Bits	Description
32	Number of Transform Points

### Wnpoints (1 Word)

Bits	Description
32	Number of Window Points

### Resolution (2 Words)

Bits	Description
44	Resolution (63..32), Hz
20	Resolution (31..0), Hz

### Span (2 Words)

Bits	Description
44	Span (63..32), Hz
20	Span (31..0), Hz

### Naverages (1 Word)

Bits	Description
32	Number of Averages

### Weighting Factor (1 Word)

Bits	Description
32	Weighting Factor

### F1 (1 Word)

Bits	Description
32	F1 index

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) Spectrum for SignalShark

---

### F2 (1 Word)

Bits	Description
32	F2 Index

### Window Time-Delta (1 Word)

Bits	Description
32	Samples

### Jumbo Packets

Jumbo packets must be supported and enabled on the SignalShark, on the control computer and on all connected LAN components like switches, routers, etc.

## 8 Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

This chapter includes information about how Vita49.2 is used with Narda SignalShark. Detailed information about Vita 49.2 can be found in the ANSI/VITA 49.2-2017 Documentation.

### IF Data Packet Class – Overview

Packet Header		
Parameter	Selected Option	Comments
Packet type	IF Data Packet with stream ID	Conveys the digitized IF
Packet Size	Variable	Depends on receiver settings.
Stream identifier	Yes	Set at run time by the user
Class ID	No	
Integer-seconds timestamp	UTC	Present in every packet
Fractional seconds timestamp	Real Time	

Packet Payload		
Parameter	Selected Option	Comments
Packing method	Processing efficient	
Item packing field size	16 or 32 bits	Specified by data packet payload field in the paired IQ Context Packet.
Data item size	16 or 32 bits	Specified by data packet payload field in the paired IQ Context Packet.
Event-tag size	0 bits	No event tags
Channel-tag size	0 bits	No channel tags
Vector size	0 bits	No vector
Real/complex type	Complex Cartesian samples	I/Q samples
Data item format	Signed fixed point	
Sample repeating/channel repeating	No repeating	No repeating of any kind
Repeat count	0	

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

Packet Trailer		
Parameter	Selected Option	Comments
Calibrated time indicator	Yes	0: timestamp is free running 1: timestamp is related to some external reference time
Valid data indicator	Yes	0: data invalid, 1: data valid Is set to 1 at detection of the “data invalid flag” OR “overflow flag” within the internal FPGA IQ data stream
Reference lock indicator	No	
AGC/MGC indicator	No	
Detected signal indicator	No	
Spectral inversion indicator	No	
Over-range indicator	Yes	0: No Over-range, 1: Over-range Is set to 1 at detection of the “overdriven flag” within the internal FPGA IQ data stream
Sample loss indicator	Yes	0: No Sample loss, 1: Sample loss Is set to 1 in the first packet after detection of a “frame number gap” > 1 within the internal FPGA IQ data stream
User-defined indicators	No	
Associated context packet count	Yes	0: No associated context packet 1: There is an associated context packet  Usually, one IQ context packet is sent during one data packet interval. Context packages are always transmitted before the associated data packet

## IF Context Packet Class – Overview

Packet Header		
Parameter	Selected Option	Comments
Packet type	IF Context Packet	
Stream identifier	Yes	Identical to the stream identifier of the associated IQ data stream.
Class ID	No	
Integer-seconds timestamp	UTC	Present in every packet
Fractional seconds timestamp	Real Time	
Timestamp precision	Packet precision	Context changes apply to the sampling interval of the data packet with the exact same timestamp (TSM = 1)



Context Fields		
Parameter	Selected Option	Comments
Context field change indicator	Yes	0: no context field changes 1: at least one context field change with respect to last conveyed packet
Reference point identifier	No	
Bandwidth	Yes	CBW (-6.02 dB BW, Parks McClellan, $\alpha = 0.16$ )
IF reference frequency	No	
RF reference frequency	Yes	Ftune
RF reference frequency offset	No	
IF band offset	No	
Reference level	Yes	Level in dBm of a CW signal at the frontend of the compensated signal chain which generates full scale I/Q data.
Gain	No	
Over-range count	No	
Sample rate	Yes	1.28×CBW(normal) or 2.56×CBW (oversampling)
Timestamp adjustment	No	
Timestamp calibration time	No	
Temperature	No	
Device identifier	No	
State and event indicators	No	Not supported. Pls. use indicator flags of the data packet instead
Data packet payload	Yes	
Logical events	No	
Calibrated time indicator	No	
Valid data indicator	No	
Reference lock indicator	No	
AGC/MGC indicator	No	
Detected signal indicator	No	
Spectral inversion indicator	No	
Over-range indicator	No	
Sample loss indicator	No	
User-defined indicators	No	
GPS	Yes	Is transmitted only when available
Formatted GPS geolocation	Yes	Altitude represents height above mean sea level
Formatted INS geolocation	No	
ECEF ephemeris	No	
Relative ephemeris	No	
GPS ASCII	No	

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

Ephemeris reference identifier	No	
Context association lists	No	
Source list	No	
System list	No	
Vector component list	No	
Asynchronous channel list	No	

## IF Data Packet Class – Details

The packet is transmitted in big-endian order, most significant byte first.

Packet Field Name	Size in Words	
Header	1	
Stream Identifier	1	
Integer-seconds Timestamp	1	
Fractional-seconds Timestamp	2	
Data Payload Available Settings depends on, CBW, Oversampling (ON/OFF) and Payload Format (16/32 Bit)	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048	
Trailer	1	

### Header (1 Word)

Bits	Value	Description
4	b0001	<b>Packet Type =</b> IF Data packet with Stream Identifier
1	0	<b>Class ID bit (C bit)</b> IF Data packets not include the Class ID field
1	1	<b>Trailer bit (T bit)</b> IF Data packets include the trailer field
2	0	<b>Reserved</b>
2	b01	<b>Timestamp integer (TSI) =</b> Coordinated Universal Time (UTC)
2	b10	<b>Timestamp fractional (TSF) =</b> Real Time (Picoseconds) Timestamp
4	0...15	<b>Packet Count</b> Modulo-16 count of IF Data packets.
16	big-endian	<b>Packet Size</b> This 16-bit field indicates the total number of 32-bit words present in the IF Data packet, including the header, payload and all other fields.

### Stream Identifier (1 Word, big-endian)

The Stream Identifier (Stream ID) is a 32-bit number assigned to a VRT Packet Stream. The Stream Identifier can be set by user and is used to separate different streams from each other.

### Integer-seconds Timestamp (1 Word, big-endian)

The Integer-seconds Timestamp conveys UTC and provides the Reference-Point Time of the first Data Sample in the packet in seconds, including leap seconds, since midnight January 1, 1970, Greenwich Mean Time.

### Fractional-seconds Timestamp (2 Words, big-endian)

The Real-Time Timestamp extends the resolution of the Integer-seconds Timestamp down to one picosecond.

### Data Payload (2...2048 Words)

The data payload is a contiguous sequence of data samples. The number of words in the data payload is variable and depends on the settings. The format of the data items within the payload is "complex cartesian" and can be selected between 16 and 32 bit resolution.

### IQ 16 Bit Payload Format (big-endian)

2x signed 16 bit fix point number

Bits	Description
1	I Sample sign bit
15	I Sample fractional part
1	Q Sample sign bit
15	Q Sample fractional part

### IQ 32 Bit Payload Format (big-endian)

2x signed 32 bit fix point number

Bits	Description
1	I Sample sign bit
24	I Sample fractional part
7	Reserved = 0
1	Q Sample sign bit
24	Q Sample fractional part
7	Reserved = 0

### Trailer (1 Word)

Bits	Value	Description
1	1	Enables Calibrated Time Indicator
1	1	Enables Valid Data Indicator
1	0	Enables Reference Lock Indicator (not implemented)
3	0	Reserved

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

1	1	Enables Over-range Indicator
1	1	Enables Sample Loss Indicator
4	0	Reserved
1	1	Calibrated Time Indicator
1	1 or 0	Valid Data Indicator
1	1 or 0	Reference Lock Indicator (not implemented)
3	0	Reserved
1	1 or 0	Over-range Indicator
1	1 or 0	Sample Loss Indicator
4	0	Reserved
1	1	Enable Associated Context Packet Count
7	1 or 0	Associated Context Packet Count

## IF Context Packet Class - Details

The packet is transmitted in big-endian order, most significant byte first.

Packet Field Name	Size in Words
Header	1
Stream Identifier	1
Integer-seconds Timestamp	1
Fractional-seconds Timestamp	2
Context Indicator Field	1
Bandwidth	2
RF Reference Frequency	2
Reference Level	1
Sample Rate	2
Data Packet Payload Format	2
Formatted GPS (optional)	11

## Header (1 Word)

Bits	Value	Description
4	b0100	<b>Packet Type</b> = IF Data context packet
1	0	<b>Class ID bit (C bit)</b> IF Data packets not include the Class ID field
2	0	<b>Reserved</b>
1	1	<b>Timestamp Mode (TSM)</b> Context changes apply to the sampling interval of the data packet
2	b01	<b>Timestamp Integer (TSI)</b> = Coordinated Universal Time (UTC)
2	b10	<b>Timestamp Fractional (TSF)</b> = Real Time (Picoseconds) Timestamp

4	0...15	<b>Packet Count</b> Modulo-16 count of IF Data packets.
16	big-endian	<b>Packet Size</b> This 16-bit field indicates the total number of 32-bit words present in the IF context packet, including the header and all other fields.

### Stream Identifier (1 Word, big-endian)

The Stream Identifier (Stream ID) is a 32-bit number assigned to a VRT Packet Stream. The Stream Identifier can be set by user and is used to separate different streams from each other.

### Integer-seconds Timestamp (1 Word, big-endian)

The Integer-seconds Timestamp conveys UTC and provides the Reference-Point Time of the first Data Sample in the packet in seconds, including leap seconds, since midnight January 1, 1970, Greenwich Mean Time.

### Fractional-seconds Timestamp (2 Words, big-endian)

The Real-Time Timestamp extends the resolution of the Integer-seconds Timestamp down to one picosecond.

### Context Indicator Field (1 Word)

The Context Indicator field contains bit fields, one for each Context Field, that indicate whether the corresponding optional Context field is present in the packet.

Bits	Value	Description
1	0 or 1	Context Field Change Indicator
1	0	Reserved
1	1	Bandwidth
1	0	Reserved
1	1	RF Reference Frequency
2	0	Reserved
1	1	Reference Level
2	0	Reserved
1	1	Sample Rate
5	0	Reserved
1	0 or 1	Data Packet Payload Format
1	0 or 1	Formatted GPS
14	0	Reserved

### Bandwidth (2 Word, big-endian)

Is an unsigned 64bit fixpoint number.

Bits	Description
44	Integer Part
20	Fractional Part

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

---

### RF Reference Frequency (2 Word, big-endian)

Is an unsigned 64bit fixpoint number.

Bits	Description
44	Integer Part
20	Fractional Part

### Reference Level (1 Word, big-endian)

32-bit format shown in the table below. The upper 16 bits of this field are reserved and should be set to zero. The Reference Level value is expressed in two's-complement format in the lower 16 bits of this field. This field has an integer and a fractional part, with the radix point to the right of bit 7.

Bits	Description
16	Reserved = 0
1	Sign bit
8	Integer Part
7	Fractional Part

### Sample Rate (2 Word, big-endian)

Is an unsigned 64bit fixpoint number.

Bits	Description
44	Integer Part
20	Fractional Part

### Data Packet Payload Format (2 Words)

Bits	Value	Description
1	0	Packing Method = processing-efficient packing
2	b01	Data Sample Type = Complex, Cartesian
5	b00000	Data Item Format = Signed Fixed-Point
1	0	Repeat Indicator = 0
3	0	Event-Tag Size = 0
4	0	Channel-Tag Size = 0
4	0	Reserved
6	b001111 or b011111	Item Packing Field Size 001111 = 16-1 for 16 Bit Format 011111 = 32-1 for 32 Bit Format
6	b001111 or b011111	Data Item Size 001111 = 16-1 for 16 Bit Format 011111 = 32-1 for 32 Bit Format
16	0	No Repeat Count
16	0	Vector Size = 0

### Formatted GPS (optional) (11 Words)

Bits	Value	Description
4	0	Reserved
2	b00	Timestamp Integer (TSI)
2	b00	Timestamp Fractional (TSF)
24	0	GPS/INS Manufacturer OUI
32	big-endian	Integer second timestamp of Position Fix
64	big-endian	Fractional second timestamp of Position Fix

## Command Reference Guide

Vita 49.2 Radio Transport (VRT) I/Q Stream for SignalShark

32	big-endian	Latitude in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Longitude in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Altitude in meters, signed 32 bit fix point number, 27 bit Integer and 5 bit fractional part
32	big-endian	Speed over Ground in meters/second, unsigned 32 bit fix point number, 16 bit Integer and 16 bit fractional part
32	big-endian	Heading Angle in degrees, (not implemented, value = 0x7FFFFFFF)
32	big-endian	Track Angle in degrees, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part
32	big-endian	Magnetic Variation, signed 32 bit fix point number, 10 bit Integer and 22 bit fractional part

## Jumbo Packets

Jumbo packets must be used for TCP and UDP for payload limit settings > 256 words to achieve maximum performance and avoid UDP packet fragmentation. The possible payload limit settings depend on CBW, Payload Format (16/32Bit) and Oversampling setting. Jumbo packets must be supported and enabled on the SignalShark, on the control computer and on all connected LAN components like switches, routers, etc.

## Glossary

Symbol	Description	
VRT	Vita Radio Transport	
Word	1 Word equals 32 bits	
Bit Range X:Z	e.g. 31:28 means bits 31,30,29,28	
Bit Value bXYZ	e.g. b1011 means 4 bits with memory content of bit1=1,bit2=0,bit3=1,bit4=1	
HexValue 0xXYZ	e.g. 0x1A equals decimal value of 26	
Big Endian	<a href="https://en.wikipedia.org/wiki/Endianness">https://en.wikipedia.org/wiki/Endianness</a>	
Fixpoint Number	<a href="https://en.wikipedia.org/wiki/Q_(number_format)">https://en.wikipedia.org/wiki/Q_(number_format)</a>	



## 9 Command Groups

### Narda Command Reference - IEEE488

<b>*IDN?</b>
Queries the instrument basic information such as manufacturer, model, serial number, version
<b>*OPC?</b>
Synchronizes overlapped commands
<b>*RST</b>
Resets the device to default settings and all status data. This command has a timeout of 10s.
<b>*WAI</b>
Synchronizes overlapped commands

### Narda Command Reference - SCPI-99

<b>ABORt</b>
Aborts running measurements and calculations. Sets TASK:STATE to STOP mode.
<b>SYSTem:ERRor[:NEXT]?</b>
Queries the error queue for the next error item and removes it from the queue. The query returns the error code and the error message string.
<b>SYSTem:ERRor:ALL?</b>
Queries the error queue for every existing error item and clears the error queue. The query returns the error code for every error.
<b>SYSTem:ERRor:CLEAr[:ALL]</b>
Clears the error queue
<b>SYSTem:ERRor:CODE[:NEXT]?</b>
Queries the error queue for the next error item and removes the error from the queue. The query returns only the error code omitting the string.
<b>SYSTem:ERRor:CODE:ALL?</b>
Queries the error queue for every existing error item and clears the error queue. The query returns the error code for every error.
<b>SYSTem:ERRor:COUNt?</b>
Queries the error queue for the number of error items
<b>SYSTem:ERRor:LIST?</b>
Queries a list of all possible error items in the error queue

### Narda Command Reference - DEvIce

<b>DEvIce:APPLication:DATE?</b>
Queries the application date
<b>DEvIce:APPLication:VERSIon?</b>
Queries the application version number
<b>DEvIce:BIOS:VERSIon?</b>
Queries the BIOS version number
<b>DEvIce:BOARd:CONTRoller:BOOTloader:VERSIon?</b>
Queries the boardcontroller bootloader version
<b>DEvIce:BOARd:CONTRoller:FIRMWare:VERSIon?</b>
Queries the boardcontroller firmware version
<b>DEvIce:FPGA:BITStream:VERSIon?</b>
Queries the FPGA bitstream version
<b>DEvIce:FPGA:PCIE:DRIVER:VERSIon?</b>
Queries the FPGA PCI Express driver version
<b>DEvIce:OPERation:SYSTem:VERSIon?</b>
Queries the operation system version
<b>DEvIce:PART:NUMBer?</b>
Queries the part number

## Command Reference Guide

### Command Groups

---

<b>DEvice:PRODUct:NAME?</b>
Queries the product name
<b>DEvice:SERIal:NUMBer?</b>
Queries the serial number

## Narda Command Reference - SETTings

<b>SETTings:ADJust:EQualizer</b>
Starts equalizer adjustment. This command has a timeout of 60s.
<b>SETTings:DATE?</b>
Queries the date setting of the system
<b>SETTings:FAST:SCANs:ONLY</b>
Allows fast scans only with limited BINS
<b>SETTings:FAST:SCANs:ONLY?</b>
Queries fast scan state
<b>SETTings:GNSS:SOURce</b>
Sets the GNSS source
<b>SETTings:GNSS:SOURce?</b>
Queries the GNSS source
<b>SETTings:MAGNetic:DECLination</b>
Sets the magnetic declination
<b>SETTings:MAGNetic:DECLination?</b>
Queries the magnetic declination
<b>SETTings:PPS:SOURce</b>
Sets the PPS source
<b>SETTings:PPS:SOURce?</b>
Queries the PPS source
<b>SETTings:REfERENCE:FREQuency:SOURce</b>
Reference frequency source
<b>SETTings:REfERENCE:FREQuency:SOURce?</b>
Queries the reference frequency source
<b>SETTings:TIME?</b>
Queries the system time?
<b>SETTings:TIME:SYNC:NOW</b>
Starts synchronization of systemtime with gnss time. This command has a timeout of 10s.
<b>SETTings:TSTamp:SYNC:MODE</b>
Sets the timestamp synchronization mode
<b>SETTings:TSTamp:SYNC:MODE?</b>
Queries the timestamp synchronization mode
<b>SETTings:TSTamp:SYNC:NOW</b>
Starts timestamp synchronization and wait until sync finished. This command has a timeout of 10s.

## Narda Command Reference - SENSE

<b>[SENSe:]ADC:OOR?</b>
Queries ADC out of range state for the active measurement task
<b>[SENSe:]ANTenna:AUTO:DATA?</b>
Queries the data of an connected auto antenna
<b>[SENSe:]ANTenna:AUTO:INFO?</b>
Queries the info data of an connected auto antenna
<b>[SENSe:]ANTenna:CHECKsum?</b>
Queries the checksum of the antenna connected to the SignalShark
<b>[SENSe:]ANTenna:HANDle?</b>
Queries the type of antenna handle connected to the SignalShark
<b>[SENSe:]ANTenna:ISOTropic:MANual</b>
Enables/Disables the manual switching of antenna axis of isotropic antennas

<b>[SENSe:]ANTenna:ISOTropic:MANual?</b>
Queries the manual switching state of antenna axis of an isotropic antenna
<b>[SENSe:]ANTenna:ISOTropic:MANual:AXIS</b>
Sets the axis which is used for measurement in case that an isotropic antenna is used
<b>[SENSe:]ANTenna:ISOTropic:MANual:AXIS?</b>
Queries which axis is used for measurement in case that an isotropic antenna is used
<b>[SENSe:]ANTenna:MANual</b>
Selects the manual antenna for the specified RF Input. An Antenna file must be edited with the built-in editor (Settings->Antennas->Create Antenna). The AntFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]ANTenna:MANual?</b>
Queries the manual antenna for the specified RF Input. An Antenna file must be edited with the built-in editor (Settings->Antennas->Create Antenna). The AntFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]ANTenna:MANual:LIST?</b>
Queries the list of available manual antennas
<b>[SENSe:]ANTenna:POLarization?</b>
Queries the polarization of the antenna connected to the antenna handle
<b>[SENSe:]ANTenna:TYPE?</b>
Queries the type of antenna connected to the SignalShark
<b>[SENSe:]ATTenuator</b>
Attenuation for the RF input
<b>[SENSe:]ATTenuator?</b>
Queries the current attenuation
<b>[SENSe:]ATTenuator:LIST?</b>
Queries a list of possible attenuator settings
<b>[SENSe:]CABLe:AUTO:DATA?</b>
Queries the data of an connected auto cable
<b>[SENSe:]CABLe:AUTO:INFO?</b>
Queries the info data of an connected auto cable
<b>[SENSe:]CABLe:CHECKsum?</b>
Queries the checksum of the cable connected to the Signal Shark
<b>[SENSe:]CABLe:MANual</b>
Selects the manual cable for the specified RF Input. A Cable file must be edited with the built-in editor (Settings->Cables->New Cable). The CableFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]CABLe:MANual?</b>
Queries the manual cable for the specified RF Input. A Cable file must be edited with the built-in editor (Settings->Cables->New Cable). The CableFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]CABLe:MANual:LIST?</b>
Queries the list of available manual cables
<b>[SENSe:]CABLe:TYPE?</b>
Queries the type of cable connected to the SignalShark
<b>[SENSe:]COMPass:DATA?</b>
Compass values of the active compass
<b>[SENSe:]EXTernal:DEvice</b>
External device
<b>[SENSe:]EXTernal:DEvice?</b>
Queries the external device state
<b>[SENSe:]EXTernal:DEvice:MANual</b>
Selects the manual external device for the specified RF Input. An External Device file must be edited with the built-in editor (Settings->Ext. Devices->New External Device). The ExtDevFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]EXTernal:DEvice:MANual?</b>
Queries the manual external device for the specified RF Input. An External Device file must be edited with the built-in editor (Settings->Ext. Devices->New External Device). The ExtDevFile Parameter equals the "Long Name" in the editor
<b>[SENSe:]EXTernal:DEvice:MANual:LIST?</b>
Queries the list of available manual external devices
<b>[SENSe:]EXTernal:DEvice:SWITChable?</b>
Queries if the external device is switchable
<b>[SENSe:]FREQuency:RANGe?</b>
Queries the frequency range for active measurement task. The frequency range depends on the connected equipment.

# Command Reference Guide

## Command Groups

<b>[SENSe:]GNSS:DATA?</b>
GNSS values of the active GNSS
<b>[SENSe:]HOLD</b>
Holds the measurement and initializes if measurement is stopped
<b>[SENSe:]HOLD?</b>
Holds the measurement and initializes if measurement is stopped
<b>[SENSe:]INPut</b>
Selects the RF input
<b>[SENSe:]INPut?</b>
Queries the RF input number
<b>[SENSe:]PREamp</b>
Preamplifier state of the antenna handle
<b>[SENSe:]PREamp?</b>
Queries the state of the preamplifier of the antenna handle
<b>[SENSe:]REference:LEVel</b>
Reference level for the RF input
<b>[SENSe:]REference:LEVel?</b>
Queries the reference level for the RF input
<b>[SENSe:]REference:LEVel:LIST?</b>
Queries a list of possible attenuator settings
<b>[SENSe:]REference:LEVel:OFFSet</b>
Sets the reference level offset. The flag [SENSe:]REference:LEVel:OFFSet:ENABLE must be enabled for the setting of [SENSe:]REference:LEVel:OFFSet to take effect!
<b>[SENSe:]REference:LEVel:OFFSet?</b>
Queries the current reference level offset. The flag [SENSe:]REference:LEVel:OFFSet:ENABLE must be enabled for the setting of [SENSe:]REference:LEVel:OFFSet to take effect!
<b>[SENSe:]REference:LEVel:OFFSet:ENABLE</b>
Enables/disables the reference level offset
<b>[SENSe:]REference:LEVel:OFFSet:ENABLE?</b>
Queries the corresponding reference level offset state
<b>[SENSe:]REference:LEVel:OFFSet:LIST?</b>
Queries a list of possible reference level offset settings
<b>[SENSe:]RUN:CONTInuous</b>
Starts a new continuous measurement
<b>[SENSe:]RUN:CONTInuous?</b>
Starts a new continuous measurement
<b>[SENSe:]RUN:CONTInuous:RESet</b>
Starts a new continuous measurement with reset
<b>[SENSe:]RUN:CONTInuous:RESet?</b>
Starts a new continuous measurement with reset
<b>[SENSe:]RUN:SINGLe</b>
Starts a new single measurement with reset
<b>[SENSe:]RUN:SINGLe?</b>
Starts a new single measurement with reset. The command processing ends when scan count equals scan number or the timeout value is reached.
<b>[SENSe:]STOP</b>
Stops the measurement
<b>[SENSe:]TSTamp:SYNC:DEVIation?</b>
Timestamp synchronization deviation
<b>[SENSe:]TSTamp:SYNC:FINE?</b>
Timestamp fine synchronization state
<b>[SENSe:]TSTamp:SYNC:STATe?</b>
Timestamp synchronization state

## Narda Command Reference - DISPlay

<b>DISPlay:LEVelmeter:LMAX</b>
Sets the maximum displayed level (x-axis) in the levelmeter view in actual unit

<b>DISPlay:LEVelmeter:LMAX?</b>
Queries the maximum displayed level (x-axis) in the levelmeter view in actual unit
<b>DISPlay:LEVelmeter:LRANge</b>
Sets the maximum displayed level range (x-axis) in the levelmeter view in actual unit
<b>DISPlay:LEVelmeter:LRANge?</b>
Queries the maximum displayed level range (x-axis) in the levelmeter view in actual unit
<b>DISPlay:MAP:CENTer</b>
Centers the map to the current GNSS position
<b>DISPlay:MAP:CENTer:POSition</b>
Sets center point of the map
<b>DISPlay:MAP:CENTer:POSition?</b>
Queries the center point of the map
<b>DISPlay:MAP:LIST?</b>
Queries a list of available maps
<b>DISPlay:MAP:SElect</b>
Sets the selected map
<b>DISPlay:MAP:SElect?</b>
Queries the name of the selected map
<b>DISPlay:MAP:ZOOM:LEVel</b>
Sets the zoom level
<b>DISPlay:MAP:ZOOM:LEVel?</b>
Queries the zoom level of the map
<b>DISPlay:PEAKtable:SHOW:TRANsmitter</b>
Shows the transmitter table with peaktable
<b>DISPlay:PEAKtable:SHOW:TRANsmitter?</b>
Queries if the transmitter table is shown with peaktable
<b>DISPlay:PEAKtable:SORT</b>
Sort criteria for peaktable
<b>DISPlay:PEAKtable:SORT?</b>
Queries the sort criteria for peaktable
<b>DISPlay:PERSiStence:LMAX</b>
Sets the maximum displayed level (y-axis) in the persistence view in actual unit
<b>DISPlay:PERSiStence:LMAX?</b>
Queries the maximum displayed level (y-axis) in the persistence view in actual unit
<b>DISPlay:PERSiStence:LRANge</b>
Sets the maximum displayed level range (y-axis) in the persistence view in actual unit
<b>DISPlay:PERSiStence:LRANge?</b>
Queries the maximum displayed level range (y-axis) in the persistence view in actual unit
<b>DISPlay:SPECTrum:LMAX</b>
Sets maximum displayed level (y-axis) in the spectrum view in actual unit
<b>DISPlay:SPECTrum:LMAX?</b>
Queries the maximum displayed level (y-axis) in the spectrum view in actual unit
<b>DISPlay:SPECTrum:LRANge</b>
Sets the maximum displayed level range (y-axis) in the spectrum view in actual unit
<b>DISPlay:SPECTrum:LRANge?</b>
Queries the maximum displayed level range (y-axis) in the spectrum view in actual unit
<b>DISPlay:UNIT</b>
Sets the actual level unit
<b>DISPlay:UNIT?</b>
Queries the actual level unit
<b>DISPlay:UNIT:LIST?</b>
Queries a list of possible level units

## Narda Command Reference - DLOGger

<b>DLOGger:SAVE:LOCalization</b>
Datalogger save localization

## Command Reference Guide

### Command Groups

<b>DLOGger:WORKing:DIRectory</b>
Datalogger get/set working directory
<b>DLOGger:WORKing:DIRectory?</b>
Datalogger get/set working directory
<b>DLOGger:CONFig:TASK:SAVE</b>
Datalogger configuration
<b>DLOGger:CONFig:TASK:SAVE?</b>
Queries the datalogger configuration
<b>DLOGger:DIRectory:ADD?</b>
Creates a new directory and returns the directory path
<b>DLOGger:DIRectory:DELeTe</b>
Deletes the given directory
<b>DLOGger:DIRectory:DELeTe:ALL</b>
Deletes all files in the given directory
<b>DLOGger:DIRectory:LIST?</b>
Queries a list of all directories within the data logger
<b>DLOGger:RECall:DATaset</b>
Recalls a dataset. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset to take effect! WARNING, this command is only recommended to be used in combination with other Querie commands. Invalid combination of commands could lead to a crash of the SignalShark application!
<b>DLOGger:RECall:DATaset?</b>
Returns the name of the current recalled dataset. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset to take effect!
<b>DLOGger:RECall:DATaset:CLOSe</b>
Closes the current dataset recall. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset:CLOSe to take effect!
<b>DLOGger:RECall:ENABled</b>
Enables/Disables recalls in remote. WARNING, this command is only recommended to be used in combination with other Querie commands. Invalid combination of commands could lead to a crash of the SignalShark application!
<b>DLOGger:RECall:ENABled?</b>
Queries the recall enabled state
<b>DLOGger:SAVE:DATaset</b>
Datalogger save dataset
<b>DLOGger:SAVE:EXTeRnal:BEARing</b>
Datalogger save external bearing
<b>DLOGger:SAVE:SCReenshot</b>
Saves screenshot to file

## Narda Command Reference - SYSTem

<b>SYSTem:AUDio:MUTE</b>
Sets the system audio mute setting
<b>SYSTem:AUDio:MUTE?</b>
Queries the system audio mute setting
<b>SYSTem:AUDio:VOLume</b>
Sets the system audio volume setting
<b>SYSTem:AUDio:VOLume?</b>
Queries the system audio volume setting
<b>SYSTem:COMMand:FILTer?</b>
Queries the filter flags of command processing
<b>SYSTem:COMMand:PROGress?</b>
Queries the progress of command processing
<b>SYSTem:MEMory:STATus?</b>
Queries the memory status of the system
<b>SYSTem:REMOte:COMMand:LIST?</b>
Queries a list with all available commands in the narda remote parser
<b>SYSTem:REMOte:DISPlay</b>
Enables or disables GUI views while remote on

<b>SYSTem:REMOte:DISPlay?</b>
Queries the remote display state
<b>SYSTem:REMOte:LOG</b>
Enables or disables remote logfile
<b>SYSTem:REMOte:LOG?</b>
Queries the remote logfile state
<b>SYSTem:REMOte:LOG:CONFig</b>
Sets the remote logfile configuration
<b>SYSTem:REMOte:LOG:CONFig?</b>
Queries the remote logfile configuration
<b>SYSTem:REMOte:SLEep?</b>
Lets parser sleep for a specific timespan and returns true if successful.
<b>SYSTem:REMOte:TIMEout</b>
SCPI default remote timeout (default factory setting = 3 seconds)
<b>SYSTem:REMOte:TIMEout?</b>
Queries the SCPI default remote timeout
<b>SYSTem:REMOte:TIMEout:RESet</b>
Resets remote timeout of all commands to the default value
<b>SYSTem:SHUTdown</b>
Shuts down or restarts the instrument
<b>SYSTem:STARtup:DATE?</b>
Startup Date of the system
<b>SYSTem:STARtup:SEConds?</b>
Seconds since last start of the system
<b>SYSTem:STARtup:TIME?</b>
Startup Time of the system

## Narda Command Reference - NETWork

<b>NETWork:MAC:ADDReSS?</b>
Queries the Network MAC address

## Narda Command Reference - TASK

<b>TASK:ADD?</b>
Adds a new measurement task to the instrument
<b>TASK:DELeTe</b>
Deletes a measurement task
<b>TASK:DELeTe:ALL</b>
Deletes all a measurement tasks
<b>TASK:LIST?</b>
Queries a list of measurement tasks and task names
<b>TASK:MOVE</b>
Moves a measurement task to another position
<b>TASK:NEW?</b>
Adds a new measurement task and deletes existing tasks. This command has a timeout of 10s
<b>TASK:REName</b>
Renames a measurement task
<b>TASK:REPLace?</b>
Replaces an existing measurement task
<b>TASK:SELeCt</b>
Selects a measurement task
<b>TASK:SELeCt?</b>
Queries the selected measurement task
<b>TASK:STATe?</b>
Queries the task state of the selected task

### Narda Command Reference - VIEW

<b>VIEW:ADD?</b>
Adds a new view to the active measurement task
<b>VIEW:DELeTe</b>
Deletes a view from the active measurement task
<b>VIEW:LIST?</b>
Queries a list of views of the active measurement task
<b>VIEW:REPLace</b>
Replaces a view of the active measurement task
<b>VIEW:SELeCt</b>
Selects a view from the active measurement task
<b>VIEW:SELeCt?</b>
Queries the index of the selected view of the active measurement task
<b>VIEW:SIZE:MAXimize</b>
Maximizes the selected view
<b>VIEW:SIZE:REStore</b>
Restores the size of the selected view

### Narda Command Reference - SPECtrum

<b>SPECTrum:DATA:ALL?</b>
Queries the spectrum configuration and values for multiple traces. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)
<b>SPECTrum:DATA:COUNt?</b>
Queries the spectrum data count (number of bins)
<b>SPECTrum:DATA:FREQuency:STARt?</b>
Queries spectrum data start frequency
<b>SPECTrum:DATA:FREQuency:STEP?</b>
Queries spectrum data frequency step
<b>SPECTrum:DATA:LEVel?</b>
Queries spectrum level data
<b>SPECTrum:DATA:OVERdriven?</b>
Queries the spectrum data overdriven flag
<b>SPECTrum:DATA:REALtime?</b>
Queries the spectrum data realtime flag for traces: MnR, AvR, MxR, MxP, MnP, MxA, MxS
<b>SPECTrum:DATA:UPDate?</b>
Update is available in RUN and HOLD mode (see DATA:UPDate Commands)
<b>SPECTrum:FREQuency:CENTer</b>
Sets the center frequency for spectrum
<b>SPECTrum:FREQuency:CENTer?</b>
Queries the center frequency
<b>SPECTrum:FREQuency:CENTer:STEP</b>
Sets the center frequency step for spectrum
<b>SPECTrum:FREQuency:CENTer:STEP?</b>
Queries the center frequency step for spectrum
<b>SPECTrum:FREQuency:CENTer:TUNE:COUPling</b>
Automatic coupling of Fcent and Ftune
<b>SPECTrum:FREQuency:CENTer:TUNE:COUPling?</b>
Queries the coupling state of Fcent and Ftune
<b>SPECTrum:FREQuency:ENTRy:MODE</b>
Sets the entry mode for Fcent, Fspan or Fstart, Fstop
<b>SPECTrum:FREQuency:ENTRy:MODE?</b>
Queries the entry mode for Fcent, Fspan or Fstart, Fstop
<b>SPECTrum:FREQuency:SPAN</b>
Frequency span for spectrum



<b>SPECTrum:FREQuency:SPAN?</b>
Queries the frequency span
<b>SPECTrum:FREQuency:START</b>
Sets the Start frequency for spectrum
<b>SPECTrum:FREQuency:START?</b>
Queries the start frequency, in Hz
<b>SPECTrum:FREQuency:STOP</b>
Sets the Stop frequency for spectrum
<b>SPECTrum:FREQuency:STOP?</b>
Queries the stop frequency, in Hz
<b>SPECTrum:MEASurement:TIME</b>
Measurement time for spectrum
<b>SPECTrum:MEASurement:TIME?</b>
Queries the measurement time
<b>SPECTrum:MEASurement:TIME:ENTRy:MODE</b>
Sets the measurement time entry mode, absolute or relative
<b>SPECTrum:MEASurement:TIME:ENTRy:MODE?</b>
Queries the measurement time entry mode, absolute or relative
<b>SPECTrum:RBW</b>
Sets the resolution bandwidth for spectrum
<b>SPECTrum:RBW?</b>
Queries the bandwidth, in Hz
<b>SPECTrum:RBW:AUTO</b>
Bandwidth automatic coupling to the span
<b>SPECTrum:RBW:AUTO?</b>
Queries the bandwidth automatic coupling state
<b>SPECTrum:RBW:ENTRy:MODE</b>
Sets the bandwidth entry mode; absolute or relative
<b>SPECTrum:RBW:ENTRy:MODE?</b>
Queries the bandwidth entry mode; absolute or relative
<b>SPECTrum:RBW:FILTer:TYPE</b>
Sets the bandwidth filter type
<b>SPECTrum:RBW:FILTer:TYPE?</b>
Queries the bandwidth filter type
<b>SPECTrum:RBW:LIST?</b>
Queries a list of possible RBW settings
<b>SPECTrum:SCAN:COUNT</b>
Sets the spectrum scan count
<b>SPECTrum:SCAN:COUNT?</b>
Queries the current scan count
<b>SPECTrum:SCAN:NUMBer?</b>
Queries the current scan number
<b>SPECTrum:TRACe:DETEctor&lt;Number&gt;</b>
Sets the detector configuration. Only the 3rd detector can be configured.
<b>SPECTrum:TRACe:DETEctor&lt;Number&gt;?</b>
Queries the detector configuration. Only the 3rd detector can be configured
<b>SPECTrum:TRACe:ENABLE</b>
Enables the displayed detectors and traces
<b>SPECTrum:TRACe:ENABLE?</b>
Queries the corresponding trace state
<b>SPECTrum:TRACe:INFinite</b>
Sets the infinite min max trace state
<b>SPECTrum:TRACe:INFinite?</b>
Queries the infinite min max trace state
<b>SPECTrum:TRACe:LIST?</b>
Queries a list of selected display traces

## Command Reference Guide

### Command Groups

#### **SPECTrum:TSTamp?**

Queries the spectrum data timestamp

## Narda Command Reference - MARKer

#### **MARKer:FXD:FREQuency**

Sets the frequency of fixed marker

#### **MARKer:FXD:FREQuency?**

Queries the frequency of fixed marker

#### **MARKer:FXD:TIME**

Sets the time position of fixed marker

#### **MARKer:FXD:TIME?**

Queries the time position of fixed marker

#### **MARKer:FXD:VALue**

Sets the value of fixed marker

#### **MARKer:FXD:VALue?**

Queries the value of fixed marker

#### **MARKer:SPECTrum:DATA:ALL?**

Queries all marker values for spectrum. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).

#### **MARKer<Index>:SPECTrum:DATA:FREQuency?**

Queries the marker or deltamarker frequency data

#### **MARKer<Index>:SPECTrum:DATA:LEVel?**

Queries marker or deltamarker level data

#### **MARKer<Index>:SPECTrum:DATA:PEAK:STATe?**

Queries the marker peak state. PEAK:STATe only works in HOLD or STOP mode or with track peak enabled (PEAK:TRACK ON.)

#### **MARKer<Index>:SPECTrum:DATA:TIME?**

Queries the marker or deltamarker time data

#### **MARKer<Index>:SPECTrum:ENABLE**

Marker enable or disable

#### **MARKer<Index>:SPECTrum:ENABLE?**

Queries the current state of a marker

#### **MARKer<Index>:SPECTrum:FREQuency**

Marker frequency (see also MARKer<Index>:SPECTrum:DATA:FREQuency?)

#### **MARKer<Index>:SPECTrum:FREQuency?**

Queries the current frequency for a marker

#### **MARKer<Index>:SPECTrum:FREQuency:LINK**

Marker frequency link

#### **MARKer<Index>:SPECTrum:FREQuency:LINK?**

Queries the marker frequency link to another marker

#### **MARKer<Index>:SPECTrum:FREQuency:LINK:OFFSet**

Marker frequency link offset

#### **MARKer<Index>:SPECTrum:FREQuency:LINK:OFFSet?**

Queries the frequency link offset

#### **MARKer<Index>:SPECTrum:FUNCTION**

Sets the marker function

#### **MARKer<Index>:SPECTrum:FUNCTION?**

Queries the channel power function

#### **MARKer:SPECTrum:FUNCTION:CPOWER:CBW**

CBW for marker function: Channel Power

#### **MARKer:SPECTrum:FUNCTION:CPOWER:CBW?**

Queries the CBW

#### **MARKer<Index>:SPECTrum:FUNCTION:CPOWER:DATA?**

Queries data for marker function: Channel Power

#### **MARKer<Index>:SPECTrum:FUNCTION:NOISe:DATA?**

Queries data for marker or deltamarker function: Noise

<b>MARKer:SPECTrum:FUNCTION:NOISe:NBW</b>
NBW for marker function: Noise
<b>MARKer:SPECTrum:FUNCTION:NOISe:NBW?</b>
Queries the NBW
<b>MARKer:SPECTrum:FUNCTION:NOISe:NBW:AUTO</b>
NBW auto coupling for marker function: Noise
<b>MARKer:SPECTrum:FUNCTION:NOISe:NBW:AUTO?</b>
Queries the NBW auto coupling state
<b>MARKer&lt;Index&gt;:SPECTrum:FUNCTION:OCBW:DATA?</b>
Queries data for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:MODE</b>
Mode for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:MODE?</b>
Mode for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:NTRials</b>
Sets the NTrials (number of trials) for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:NTRials?</b>
NTrials (number of trials) for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:PERCent</b>
Sets the % for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:PERCent?</b>
Queries the % for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:THReshold</b>
Sets the threshold for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:THReshold?</b>
Queries the threshold for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:XDB</b>
XdB for marker function: OCCUPIED_BANDWIDTH
<b>MARKer:SPECTrum:FUNCTION:OCBW:XDB?</b>
XdB for marker function: OCCUPIED_BANDWIDTH
<b>MARKer&lt;Index&gt;:SPECTrum:FUNCTION:TRANsmitter:DATA?</b>
Queries the data for Marker function: TRANSMITTER
<b>MARKer:SPECTrum:LIST?</b>
List of enabled markers
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK</b>
Marker peak search (time or frequency domain)
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK:LEFT</b>
Marker left peak search (frequency domain)
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK:LOWer</b>
Marker lower peak search (time domain)
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK:NEXT</b>
Marker next peak search (time or frequency domain)
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK:RIGHT</b>
Marker right peak search (frequency domain)
<b>MARKer&lt;Index&gt;:SPECTrum:PEAK:UPPer</b>
Marker upper peak search (time domain)
<b>MARKer&lt;Index&gt;:SPECTrum:REFerence</b>
Reference marker for a delta marker
<b>MARKer&lt;Index&gt;:SPECTrum:REFerence?</b>
Queries the corresponding reference marker
<b>MARKer:SPECTrum:SEARch:AUTO:PEAK:ENABLE</b>
'Auto Peak Search for M1' for marker
<b>MARKer:SPECTrum:SEARch:AUTO:PEAK:ENABLE?</b>
Queries the searchlimit enable state
<b>MARKer:SPECTrum:SEARch:FREQuency:LOWer</b>
Lower frequency searchlimit for marker and peaktable

# Command Reference Guide

## Command Groups

<b>MARKer:SPECTrum:SEARch:FREQuency:LOWer?</b>
Queries the lower frequency searchlimit
<b>MARKer:SPECTrum:SEARch:FREQuency:UPPer</b>
Upper frequency searchlimit for marker and peaktable
<b>MARKer:SPECTrum:SEARch:FREQuency:UPPer?</b>
Queries the upper frequency searchlimit
<b>MARKer:SPECTrum:SEARch:LIMits:ENABle</b>
Sets the 'Use Search Limits' enable state for marker and peaktable
<b>MARKer:SPECTrum:SEARch:LIMits:ENABle?</b>
Queries the searchlimit enable state
<b>MARKer:SPECTrum:SEARch:LOEXclude:ENABle</b>
'Exclude LO' searchlimit for marker and peaktable
<b>MARKer:SPECTrum:SEARch:LOEXclude:ENABle?</b>
Queries the searchlimit enable state
<b>MARKer:SPECTrum:SEARch:PEAK:EXCursion</b>
Sets the peak excursion for marker and peaktable
<b>MARKer:SPECTrum:SEARch:PEAK:EXCursion?</b>
Queries the peak excursion
<b>MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABle</b>
'Peak Excursion' searchlimit for marker and peaktable
<b>MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABle?</b>
Queries the peak excursion enable state
<b>MARKer:SPECTrum:SEARch:SGRam:RANge</b>
Sets the spectrogram searchrange
<b>MARKer:SPECTrum:SEARch:SGRam:RANge?</b>
Queries the spectrogram searchrange
<b>MARKer:SPECTrum:SEARch:THReshold</b>
Threshold searchlimit for marker and peaktable
<b>MARKer:SPECTrum:SEARch:THReshold?</b>
Queries the threshold searchlimit
<b>MARKer:SPECTrum:SEARch:TRACk:PEAKs:ENABle</b>
'Track Peaks' for marker
<b>MARKer:SPECTrum:SEARch:TRACk:PEAKs:ENABle?</b>
Queries the searchlimit enable state
<b>MARKer&lt;Index&gt;:SPECTrum:TIME</b>
Time position of a marker (see also MARKer<Index>:SPECTrum:DATA:TIME?)
<b>MARKer&lt;Index&gt;:SPECTrum:TIME?</b>
Queries marker time position
<b>MARKer&lt;Index&gt;:SPECTrum:TIME:LINK</b>
Marker time link
<b>MARKer&lt;Index&gt;:SPECTrum:TIME:LINK?</b>
Queries the time link marker
<b>MARKer&lt;Index&gt;:SPECTrum:TIME:LINK:OFFSet</b>
Marker time link offset. A time offset can only be set if the selected task is an RT Spectrum task.
<b>MARKer&lt;Index&gt;:SPECTrum:TIME:LINK:OFFSet?</b>
Queries the time link offset
<b>MARKer&lt;Index&gt;:SPECTrum:TRACe</b>
Selects the active trace for a marker. Only traces enabled by SPECTrum:TRACe:ENABle are available.
<b>MARKer&lt;Index&gt;:SPECTrum:TRACe?</b>
Queries the active trace of a marker
<b>MARKer&lt;Index&gt;:SPECTrum:TYPE</b>
Sets the Marker type
<b>MARKer&lt;Index&gt;:SPECTrum:TYPE?</b>
Queries the configuration of a marker, normal or delta

## Narda Command Reference - LEVelmeter

<b>LEVelmeter:CBW</b>
Sets the channel bandwidth for levelmeter
<b>LEVelmeter:CBW?</b>
Queries the channel bandwidth for levelmeter
<b>LEVelmeter:CBW:FILTer:TYPE</b>
Sets the levelmeter CBW filter type
<b>LEVelmeter:CBW:FILTer:TYPE?</b>
Queries the levelmeter CBW filter type
<b>LEVelmeter:CBW:LIST?</b>
Queries a list of possible CBW settings
<b>LEVelmeter:CBW:OVERsampling</b>
Sets the levelmeter oversampling state
<b>LEVelmeter:CBW:OVERsampling?</b>
Queries the levelmeter oversampling state
<b>LEVelmeter:DATA:AFC?</b>
Queries AFC data values
<b>LEVelmeter:DATA:ALL?</b>
Queries all levelmeter detector and handle values. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).
<b>LEVelmeter:DATA:DETEctor&lt;Number&gt;?</b>
Queries the levelmeter detector value (PPk, CPk, RMS, CRMS, MPk, Smp, Avg, CAvg)
<b>LEVelmeter:DATA:DETEctor:MODulation?</b>
Queries the levelmeter modulation detector values (AM_PPk, AM_MPk, AM_PPDiv2, AM_RMS, FM_PPk, FM_MPk, FM_PPDiv2, FM_RMS, PM_PPk, PM_MPk, PM_PPDiv2, PM_RMS)
<b>LEVelmeter:DATA:UPDate?</b>
Update is available in RUN and HOLD mode (see DATA:UPDate Commands)
<b>LEVelmeter:DETEctor&lt;Number&gt;</b>
Detector configuration
<b>LEVelmeter:DETEctor&lt;Number&gt;?</b>
Queries the detector configuration
<b>LEVelmeter:DETEctor:INFinite</b>
Sets the infinite min max detector state
<b>LEVelmeter:DETEctor:INFinite?</b>
Queries the infinite min max detector state
<b>LEVelmeter:DETEctor:MODulation</b>
Modulation detector configuration
<b>LEVelmeter:DETEctor:MODulation?</b>
Queries the modulation detector configuration
<b>LEVelmeter:FREQuency:TUNE</b>
Sets the levelmeter tune frequency
<b>LEVelmeter:FREQuency:TUNE?</b>
Queries the levelmeter tune frequency
<b>LEVelmeter:FREQuency:TUNE:CENTer:COUPling</b>
Automatic coupling of Ftune and Fcent
<b>LEVelmeter:FREQuency:TUNE:CENTer:COUPling?</b>
Queries the coupling state of Ftune and Fcent
<b>LEVelmeter:FREQuency:TUNE:STEP</b>
Tune frequency step for levelmeter
<b>LEVelmeter:FREQuency:TUNE:STEP?</b>
Queries the tune frequency step for levelmeter
<b>LEVelmeter:MEASurement:TIME</b>
Sets the measurement time for levelmeter
<b>LEVelmeter:MEASurement:TIME?</b>
Queries the measurement time for levelmeter

## Command Reference Guide

### Command Groups

---

<b>LEVelmeter:POST:AVG</b>
Sets the post averaging time for levelmeter
<b>LEVelmeter:POST:AVG?</b>
Queries the post averaging time
<b>LEVelmeter:PRE:AVG</b>
Sets the pre averaging time for levelmeter
<b>LEVelmeter:PRE:AVG?</b>
Queries the pre averaging time
<b>LEVelmeter:SCAN:COUNT</b>
Sets the levelmeter scan count
<b>LEVelmeter:SCAN:COUNT?</b>
Queries the levelmeter scan count
<b>LEVelmeter:SCAN:NUMBER?</b>
Queries the levelmeter scan number
<b>LEVelmeter:TSTamp?</b>
Queries levelmeter data timestamp

## Narda Command Reference - PEAKtable

<b>PEAKtable:DATA:ALL?</b>
Queries peaktable with frequency and level values for multiple traces. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).
<b>PEAKtable:DATA:COUNT?</b>
Queries the number of peaktable list elements
<b>PEAKtable:DATA:FREQuency?</b>
Queries peaktable frequency data
<b>PEAKtable:DATA:LEVel?</b>
Queries peaktable level data
<b>PEAKtable:DATA:SCAN:NUMBER?</b>
PEAKtable data scan number
<b>PEAKtable:DATA:TSTamp?</b>
Query PEAKtable data timestamp
<b>PEAKtable:DATA:UPDate?</b>
Update is available in RUN and HOLD mode (see DATA:UPDate Commands)
<b>PEAKtable:SCAN:NUMBER?</b>
Queries the peaktable scan number
<b>PEAKtable:TRACe</b>
Selects the active trace for the peaktable. Only traces enabled by SPECTrum:TRACe:ENABLE are available.
<b>PEAKtable:TRACe?</b>
Queries the active trace
<b>PEAKtable:TSTamp?</b>
Queries peaktable data timestamp

## Narda Command Reference - STReam

<b>STReam:ADD?</b>
Adds a new stream to the active stream task
<b>STReam:CONNECTION:ADDReSS</b>
Sets the connection IP or multicast address for selected stream
<b>STReam:CONNECTION:ADDReSS?</b>
Queries the stream IP address
<b>STReam:CONNECTION:CLOSe</b>
Closes connection for selected stream
<b>STReam:CONNECTION:IDN</b>
Stream identifier for the selected stream connection
<b>STReam:CONNECTION:IDN?</b>
Queries the stream identifier of the stream connection

<b>STReam:CONNection:OPEN</b>
Opens connection for selected stream
<b>STReam:CONNection:PORT</b>
Connection portnumber for selected stream
<b>STReam:CONNection:PORT?</b>
Queries the connection portnumber for selected stream
<b>STReam:CONNection:STATe?</b>
Queries connection state for selected stream
<b>STReam:CONNection:TYPE</b>
Sets the connection type TCP or UDP for selected stream
<b>STReam:CONNection:TYPE?</b>
Queries the connection type for selected stream
<b>STReam:DELeTe</b>
Deletes a stream from the active stream task
<b>STReam:LIST?</b>
Queries a list of streams from the active stream task
<b>STReam:SELeCt</b>
Selects a stream of the active stream task for configuration of the connection
<b>STReam:SELeCt?</b>
Queries the selected stream of the active stream task

## Narda Command Reference - IQSTream

<b>IQSTream:ANTenna:FACTor?</b>
Queries the antenna factor for Ftune
<b>IQSTream:CBW</b>
Sets the IQ channel bandwidth
<b>IQSTream:CBW?</b>
Queries the IQ channel bandwidth
<b>IQSTream:CBW:LIST?</b>
Queries a list of possible CBW settings
<b>IQSTream:FREQuency:TUNE</b>
Sets the IQ tune frequency
<b>IQSTream:FREQuency:TUNE?</b>
Queries the IQ tune frequency
<b>IQSTream:OVERsampling</b>
Sets the IQ oversampling state
<b>IQSTream:OVERsampling?</b>
Queries the IQ oversampling state
<b>IQSTream:PAYLoad:FORMat</b>
Sets the IQ payload format
<b>IQSTream:PAYLoad:FORMat?</b>
Queries the IQ payload format
<b>IQSTream:PAYLoad:LIMit</b>
Sets the IQ payload limit in words (32Bit)
<b>IQSTream:PAYLoad:LIMit?</b>
Queries the IQ payload limit
<b>IQSTream:PAYLoad:LIMit:LIST?</b>
Queries a list of possible payload limits depending on payload format, CBW and oversampling

## Narda Command Reference - BEARing

<b>BEARing:AZIMuth:CORRection</b>
Sets the azimuth correction for bearing
<b>BEARing:AZIMuth:CORRection?</b>
Queries the azimuth correction for bearing

# Command Reference Guide

## Command Groups

<b>BEARing:CBW</b>
Sets the channel bandwidth for bearing
<b>BEARing:CBW?</b>
Queries the channel bandwidth for bearing
<b>BEARing:CBW:LIST?</b>
Queries a list of possible CBW settings
<b>BEARing:CYCLE:TIME?</b>
Queries the cycle time for bearing
<b>BEARing:DATA:ALL?</b>
Queries all bearing values. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)
<b>BEARing:DATA:AZIMuth?</b>
Queries the azimuth of the bearing result.
<b>BEARing:DATA:DETECTOR?</b>
Queries the bearing detector value
<b>BEARing:DATA:DFQuality?</b>
Queries the DF quality of the current bearing in percent
<b>BEARing:DATA:ELEVation?</b>
Queries the elevation of the bearing result.
<b>BEARing:DATA:SCAN:NUMBER?</b>
Queries the bearing data scan number
<b>BEARing:DATA:TSTamp?</b>
Queries the bearing data timestamp
<b>BEARing:DATA:UPDATE?</b>
Update is available in RUN and HOLD mode (see DATA:UPDATE Commands)
<b>BEARing:DATA:VALID?</b>
Queries the valid flag of the last bearing result
<b>BEARing:DATA:VALID:ENABLE</b>
Switches 'Last valid bearing data' flag on/off. When enabled, last valid bearing is returned and not the last measurement run data. Filter settings like 'BEARing:DFSquelch', 'BEARing:MIN:DFQuality' or 'BEARing:MIN:STABILITY' will only take effect, if this parameter is enabled!
<b>BEARing:DATA:VALID:ENABLE?</b>
Queries last valid bearing data state. When enabled, last valid bearing is returned and not the last measurement run data. Filter settings like 'BEARing:DFSquelch', 'BEARing:MIN:DFQuality' or 'BEARing:MIN:STABILITY' will only take effect, if this parameter is enabled!
<b>BEARing:DFCorrection:LIST?</b>
Queries a list of DF correction files
<b>BEARing:DFCorrection:SELEct</b>
Selects the DF correction file. A DFCorrection file must be uploaded to the SignalShark and saved under D:\Narda_SignalShark\Configuration\DFCorrection. DFCorrFile is the name of the .xml-file thus for MyFile.xml is the DFCorrFile parameter "MyFile". Note: It is only possible to select or query a DF correction file when an Auto DF-task is selected (see TASK:SELEct and TASK:SELEct? commandos to select a task and query the selected task)
<b>BEARing:DFCorrection:SELEct?</b>
Queries the selected DF correction file. A DFCorrection file must be uploaded to the SignalShark and saved under D:\Narda_SignalShark\Configuration\DFCorrection. DFCorrFile is the name of the .xml-file thus for MyFile.xml is the DFCorrFile parameter "MyFile". Note: It is only possible to select or query a DF correction file when an Auto DF-task is selected (see TASK:SELEct and TASK:SELEct? commandos to select a task and query the selected task)
<b>BEARing:DFSquelch</b>
Sets the DF squelch for bearing. The flags BEARing:DFSquelch:ENABLE and BEARing:DATA:VALID:ENABLE must be enabled for the settings of BEARing:DFSquelch to take effect!
<b>BEARing:DFSquelch?</b>
Queries the DF squelch level for bearing. The flags BEARing:DFSquelch:ENABLE and BEARing:DATA:VALID:ENABLE must be enabled for the settings of BEARing:DFSquelch to take effect!
<b>BEARing:DFSquelch:ENABLE</b>
Enables/Disables DF squelch criteria for bearings. The flag BEARing:DATA:VALID:ENABLE must also be enabled for the settings of BEARing:DFSquelch:ENABLE to take effect.
<b>BEARing:DFSquelch:ENABLE?</b>
Queries the DF squelch state. The flag BEARing:DATA:VALID:ENABLE must also be enabled for the settings of BEARing:DFSquelch:ENABLE to take effect.



<b>BEARing:FREQuency:TUNE</b>
Sets the bearing tune frequency
<b>BEARing:FREQuency:TUNE?</b>
Queries the bearing tune frequency
<b>BEARing:FREQuency:TUNE:STEP</b>
Sets the tune frequency step for bearing
<b>BEARing:FREQuency:TUNE:STEP?</b>
Queries the tune frequency step for bearing
<b>BEARing:MEASurement:TIME</b>
Sets the measurement time for bearing
<b>BEARing:MEASurement:TIME?</b>
Queries the measurement time for bearing
<b>BEARing:MIN:DFQuality</b>
Sets the minimum DF quality for a bearing. The flags BEARing:MIN:DFQuality:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:MIN:DFQuality to take effect!
<b>BEARing:MIN:DFQuality?</b>
Queries the minimum DF quality for a bearing. Sets the minimum DF quality for a bearing. The flags BEARing:MIN:DFQuality:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:MIN:DFQuality to take effect!
<b>BEARing:MIN:DFQuality:ENABLE</b>
Enables/Disables minimum DF quality criteria for bearings. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:DFQuality:ENABLE to take effect
<b>BEARing:MIN:DFQuality:ENABLE?</b>
Queries the minimum DF quality criteria state. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:DFQuality:ENABLE to take effect.
<b>BEARing:MIN:STABILITY</b>
Sets the minimum stability for bearings. The flags BEARing:DATA:VALid:ENABLE and BEARing:MIN:STABILITY:ENABLE must be enabled for the settings of BEARing:MIN:STABILITY to take effect!
<b>BEARing:MIN:STABILITY?</b>
Queries the minimum level stability for bearings. The flags BEARing:DATA:VALid:ENABLE and BEARing:MIN:STABILITY:ENABLE must be enabled for the settings of BEARing:MIN:STABILITY to take effect!
<b>BEARing:MIN:STABILITY:ENABLE</b>
Enables/Disables minimum stability criteria for bearings. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:STABILITY:ENABLE to take effect.
<b>BEARing:MIN:STABILITY:ENABLE?</b>
Queries the minimum stability criteria state. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:STABILITY:ENABLE to take effect.
<b>BEARing:NORTH:REference</b>
Sets the north reference for bearing
<b>BEARing:NORTH:REference?</b>
Queries the north reference for bearing
<b>BEARing:POST:AVG</b>
Sets the post averaging time for bearing
<b>BEARing:POST:AVG?</b>
Queries the post averaging time for bearing
<b>BEARing:REference:MARK:DIRection</b>
Sets the reference mark direction for bearing
<b>BEARing:REference:MARK:DIRection?</b>
Queries the reference mark direction for bearing
<b>BEARing:SCAN:COUNt</b>
Sets the bearing scan count
<b>BEARing:SCAN:COUNt?</b>
Queries the bearing scan count
<b>BEARing:SCAN:NUMBer?</b>
Queries the bearing scan number
<b>BEARing:SCAN:TIME?</b>
Queries the scan time for bearing
<b>BEARing:TSTamp?</b>
Queries the bearing data timestamp

### Narda Command Reference - DEMod

<b>DEMod:CBW</b>
Sets the channel bandwidth for demodulation
<b>DEMod:CBW?</b>
Queries the channel bandwidth for demodulation
<b>DEMod:CBW:FILTer:TYPE</b>
Sets the demodulation CBW filter type
<b>DEMod:CBW:FILTer:TYPE?</b>
Queries the demodulation CBW filter type
<b>DEMod:CBW:LIST?</b>
Queries a list of possible CBW settings
<b>DEMod:CBW:OVERsampling</b>
Sets the demodulation oversampling state
<b>DEMod:CBW:OVERsampling?</b>
Queries the demodulation oversampling state
<b>DEMod:FREQuency:TUNE</b>
Sets the demodulation tune frequency
<b>DEMod:FREQuency:TUNE?</b>
Queries the demodulation tune frequency
<b>DEMod:FREQuency:TUNE:AFC</b>
Switches the demodulation AFC on/off
<b>DEMod:FREQuency:TUNE:AFC?</b>
Queries the demodulation AFC state
<b>DEMod:FREQuency:TUNE:BFO</b>
Sets the demodulation BFO value (only for USB,ISB,LSB)
<b>DEMod:FREQuency:TUNE:BFO?</b>
Queries the demodulation BFO value (only for USB,ISB,LSB)
<b>DEMod:MUTE</b>
Sets the mute state of the demodulation volume
<b>DEMod:MUTE?</b>
Queries the mute state of the demodulation volume
<b>DEMod:SQUelch</b>
Sets the demodulation squelch value. The command DEMod:SQUelch:ENABLE must be enabled for the settings of DEMod:SQUelch to take effect!
<b>DEMod:SQUelch?</b>
Queries the demodulation squelch value. The command DEMod:SQUelch:ENABLE must be enabled for the settings of DEMod:SQUelch to take effect!
<b>DEMod:SQUelch:ENABLE</b>
Switches the demodulation squelch on/off
<b>DEMod:SQUelch:ENABLE?</b>
Queries the demodulation squelch enable state
<b>DEMod:TYPE</b>
Sets the demodulation type
<b>DEMod:TYPE?</b>
Queries the demodulation type
<b>DEMod:TYPE:ENABLE</b>
Switches the demodulation type on/off
<b>DEMod:TYPE:ENABLE?</b>
Queries the demodulation type enable state
<b>DEMod:VOLume</b>
Sets the demodulation volume
<b>DEMod:VOLume?</b>
Queries the demodulation volume
<b>DEMod:VOLume:AGC</b>
Switches the demodulation AGC on/off

**DEMod:VOLume:AGC?**

Queries the demodulation AGC state

## Narda Command Reference - SGRam

**SGRam:DATA:ALL?**

Queries all or a range of spectrogram frames. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)

**SGRam:DATA:FRAMe:COUNT?**

Queries the number of valid spectrogram frames

**SGRam:DATA:UPDate?**

Update is available in RUN and HOLD state. (see DATA:UPDate Commands)

**SGRam:DETEctor<Number>**

Sets the spectrogram detector configuration

**SGRam:DETEctor<Number>?**

Queries the spectrogram detector configuration

**SGRam:DETEctor:VISible**

Sets the spectrogram detector configuration

**SGRam:DETEctor:VISible?**

Queries the spectrogram detector configuration

**SGRam:FRAMe:COUNT**

Sets the maximum number of spectrogram frames

**SGRam:FRAMe:COUNT?**

Queries the maximum number of spectrogram frames

**SGRam:SCAN:NUMBer?**

Queries the spectrogram scan number

## Narda Command Reference - MAP

**MAP:LOCalization:AREA**

Sets the localization area.

**MAP:LOCalization:AREA?**

Queries the localization area.

**MAP:LOCalization:BEARing:ERRor**

Sets the bearing error for the localization

**MAP:LOCalization:BEARing:ERRor?**

Queries the bearing error for the localization

**MAP:LOCalization:DATA?**

Queries the localization data

**MAP:LOCalization:FREQuency:TUNE:LIST?**

Queries the available Ftunes for localization

**MAP:LOCalization:FREQuency:TUNE:SElect**

Sets the frequency of the localization

**MAP:LOCalization:FREQuency:TUNE:SElect?**

Queries the frequency of the localization

**MAP:LOCalization:LOS:PROBability**

Sets the line of sight probability for the localization

**MAP:LOCalization:LOS:PROBability?**

Queries the line of sight probability for the localization

**MAP:LOCalization:MIN:DFQuality**

Sets the minimum DF quality for a bearing. The flag MAP:LOCalization:MIN:DFQuality:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:DFQuality to take effect!

**MAP:LOCalization:MIN:DFQuality?**

Queries the minimum DF quality for a bearing to be taken into account by the localization. The flag MAP:LOCalization:MIN:DFQuality:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:DFQuality to take effect!

**MAP:LOCalization:MIN:DFQuality:ENABLE**

Enables/Disables minimum DF quality criteria for bearings.

## Command Reference Guide

### Command Groups

<b>MAP:LOCalization:MIN:DFQuality:ENABLE?</b>
Queries the minimum DF quality criteria state.
<b>MAP:LOCalization:MIN:SPEed</b>
Sets the min. Speed parameter for the localization. The flag MAP:LOCalization:MIN:SPEed:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:SPEed to take effect!
<b>MAP:LOCalization:MIN:SPEed?</b>
Queries the min. Speed parameter for the localization. The flag MAP:LOCalization:MIN:SPEed:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:SPEed to take effect!
<b>MAP:LOCalization:MIN:SPEed:ENABLE</b>
Enables/Disables minimum speed criteria for bearings to be taken into account by the localization.
<b>MAP:LOCalization:MIN:SPEed:ENABLE?</b>
Queries the minimum speed criteria state for bearings to be taken into account by the localization.
<b>MAP:LOCalization:MIN:SPEed:LIST?</b>
Queries the list of possible values for the min. Speed parameter
<b>MAP:LOCalization:RECORD:BEARING</b>
Starts/Stops recording of bearings
<b>MAP:LOCalization:RECORD:BEARING?</b>
Queries the recording of bearings state
<b>MAP:LOCalization:RECORD:LIST?</b>
Queries the available records for localization
<b>MAP:LOCalization:RECORD:SELEct</b>
Selects the record used for localization
<b>MAP:LOCalization:RECORD:SELEct?</b>
Queries the record used for localization
<b>MAP:LOCalization:RESolution</b>
Sets the resolution of the localization area
<b>MAP:LOCalization:RESolution?</b>
Queries the resolution of the localization area
<b>MAP:LOCalization:START</b>
Starts the localization
<b>MAP:LOCalization:STATe?</b>
Queries the localization state
<b>MAP:LOCalization:STOP</b>
Stops the localization
<b>MAP:LOCalization:USE:CASE:PRESet</b>
Sets the use case preset for the localization
<b>MAP:LOCalization:USE:CASE:PRESet?</b>
Queries the use case preset for the localization
<b>MAP:MODE</b>
Sets the map mode
<b>MAP:MODE?</b>
Queries the map mode

## Narda Command Reference - HORizontal

<b>HORizontal:SCAN:ADD:VALue</b>
Horizontal scan add discrete value
<b>HORizontal:SCAN:CALC:RESult</b>
Horizontal scan calculate bearing
<b>HORizontal:SCAN:DATA?</b>
Queries the Horizontal scan results
<b>HORizontal:SCAN:DELEte:VALue</b>
Horizontal scan delete last value
<b>HORizontal:SCAN:DETEctor</b>
Sets the Horizontal scan detector
<b>HORizontal:SCAN:DETEctor?</b>
Queries the Horizontal scan detector

<b>HORizontal:SCAN:MANual:CORRection</b>
Horizontal scan manual correction
<b>HORizontal:SCAN:RESet</b>
Horizontal scan reset
<b>HORizontal:SCAN:RESet:MAX</b>
Horizontal scan reset max
<b>HORizontal:SCAN:STARt</b>
Horizontal scan start
<b>HORizontal:SCAN:STATe?</b>
Queries the Horizontal scan state
<b>HORizontal:SCAN:STOP</b>
Horizontal scan stop
<b>HORizontal:SCAN:TYPE</b>
Sets the Horizontal scan type
<b>HORizontal:SCAN:TYPE?</b>
Queries the Horizontal scan type

## Narda Command Reference - PERSistence

<b>PERSistence:DATA:ALL?</b>
Queries a persistence image. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)
<b>PERSistence:DATA:COLUMN:COUNT?</b>
Queries the number of columns of a persistence image
<b>PERSistence:DATA:FREQUENCY:STARt?</b>
Queries persistence data start frequency
<b>PERSistence:DATA:FREQUENCY:STEP?</b>
Queries persistence data frequency step
<b>PERSistence:DATA:OVERdriven?</b>
Queries the persistence data overdriven flag
<b>PERSistence:DATA:REALtime?</b>
Queries the persistence data realtime flag
<b>PERSistence:DATA:ROW:COUNT?</b>
Queries the number of rows of a persistence image
<b>PERSistence:DATA:UPDate?</b>
Update is available in RUN and HOLD mode (see DATA:UPDate Commands)
<b>PERSistence:MEASurement:TIME</b>
Measurement time for persistence
<b>PERSistence:MEASurement:TIME?</b>
Queries the persistence measurement time
<b>PERSistence:SCAN:COUNT</b>
Sets the persistence scan count
<b>PERSistence:SCAN:COUNT?</b>
Queries the current scan count
<b>PERSistence:SCAN:NUMBER?</b>
Queries the persistence scan number
<b>PERSistence:TYPE</b>
Sets the persistence type
<b>PERSistence:TYPE?</b>
Queries the persistence type

## Narda Command Reference - Deprecated

<b>BEARing:PERSistence</b>
Deprecated Command: Use BEARing:POST:AVG instead
<b>BEARing:PERSistence?</b>
Deprecated Command: Use BEARing:POST:AVG? instead

## Command Reference Guide

### Command Groups

---

<b>LEVelmeter:DETECTOR:PERSistence</b>
Deprecated Command: Use LEVelmeter:POST:AVG instead
<b>LEVelmeter:DETECTOR:PERSistence?</b>
Deprecated Command: Use LEVelmeter:POST:AVG? instead
<b>LEVelmeter:TVIDeo</b>
Deprecated Command: Use LEVelmeter:PRE:AVG instead
<b>LEVelmeter:TVIDeo?</b>
Deprecated Command: Use LEVelmeter:PRE:AVG? instead
<b>[SENSe:]APPLication:QUIT</b>
Deprecated Command: Use SYSTem:SHUTdow instead
<b>[SENSe:]ATTenuator:ENTRy:MODE</b>
Deprecated Command: Has no effect anymore
<b>[SENSe:]ATTenuator:ENTRy:MODE?</b>
Deprecated Command: Has no effect anymore
<b>[SENSe:]REFerence:LEVel:ENTRy:MODE</b>
Deprecated Command: Has no effect anymore
<b>[SENSe:]REFerence:LEVel:ENTRy:MODE?</b>
Deprecated Command: Has no effect anymore
<b>[SENSe:]RUN:SINGle:OVERlap</b>
Deprecated Command: Use RUN:SINGle instead
<b>[SENSe:]STOP:MODE</b>
Deprecated Command: Use STOP and HOLD instead
<b>[SENSe:]STOP:MODE?</b>
Deprecated Command: Use STOP and HOLD instead

## 10 Command Descriptions

### Narda Command Reference - IEEE488

Contains the SCPI commands as defined in the IEEE488.2

#### \*IDN?

Queries the instrument basic information such as manufacturer, model, serial number, version

#### Result Parameter

Parameter	Type	Description
Identification	Arbitrary[String]	Device identification string

#### Examples

//Queries the instrument basic information

\*IDN?

Narda Safety Test Solutions GmbH,SignalShark 3310,A-0054,V1.3.1

#### \*OPC?

Synchronizes overlapped commands

#### Result Parameter

Parameter	Type	Description
OPC_Flag	UShort	OPC Flag (will always be 1)

#### Examples

//Waits until all operations are complete and then returns 1

\*OPC?

1

#### \*RST

Resets the device to default settings and all status data. This command has a timeout of 10s.

#### Examples

//Resets the device

\*RST

#### \*WAI

Synchronizes overlapped commands

### Examples

//Waits until all commands are finished

\*WAI



## Narda Command Reference - SCPI-99

Scpi commands as defined in the Scpi99 standard.

### ABORt

Aborts running measurements and calculations. Sets TASK:STATE to STOP mode.

#### Examples

```
//Aborts any running measurement
```

```
ABORt
```

### SYSTem:ERRor[:NEXT]?

Queries the error queue for the next error item and removes it from the queue. The query returns the error code and the error message string.

#### Result Parameter

Parameter	Type	Description
ErrorCode	Long	Next error code from the queue
ErrorMessage	String	Next error string from the queue

#### Examples

```
//Queries the next error code and string from the error queue
```

```
SYST:ERR?
```

```
-113,"Undefined header :SYSTem:ERRor:CODE:NEXT?"
```

### SYSTem:ERRor:ALL?

Queries the error queue for every existing error item and clears the error queue. The query returns the error code for every error.

#### Result Parameter

Parameter	Type	Description
ErrorCodes	FlexArray[Long,String]	Array of all error codes and error strings

#### Examples

```
//Queries a list of all error items in the error queue
```

```
SYST:ERR:ALL?
```

```
-100,"Command error :SPEC:FREQ:STOP",-224,"eERR_ILLEGAL_PARAMETER_VALUE"
```

### SYSTem:ERRor:CLEar[:ALL]

Clears the error queue

#### Examples

```
//Clears the error queue
```

```
SYSTem:ERRor:CLEar
```

### SYSTem:ERRor:CODE[:NEXT]?

Queries the error queue for the next error item and removes the error from the queue. The query returns only the error code omitting the string.

#### Result Parameter

Parameter	Type	Description
ErrorCode	Long	Next error code from the queue

#### Examples

//Queries the next error code from the error queue

SYST:ERR:CODE?

-113

### SYSTem:ERRor:CODE:ALL?

Queries the error queue for every existing error item and clears the error queue. The query returns the error code for every error.

#### Result Parameter

Parameter	Type	Description
ErrorCodes	FlexArray[Long]	Array of all error codes

#### Examples

//Queries a list of every error in the error queue

SYSTem:ERRor:CODE:ALL?

-113,-113,-113

### SYSTem:ERRor:COUNt?

Queries the error queue for the number of error items

#### Result Parameter

Parameter	Type	Description
ErrorCount	ULongLong	Count of errors in the error queue

#### Examples

//Queries the current size of the error queue

SYST:ERR:COUN?

2

### SYSTem:ERRor:LIST?

Queries a list of all possible error items in the error queue

**Result Parameter**

Parameter	Type	Description
ErrorCodes	FlexArray[Long,String]	Array of all error codes and error strings

**Examples**

//Queries a list of all possible error items in the error queue

SYSTem:ERRor:LIST?

-440,"eERR\_QUERY\_UNTERMINATED\_AFTER\_INDEFINITE\_RESPONSE", ...

## Narda Command Reference - DEvice

Device subsystem

### DEvice:APPLication:DATE?

Queries the application date

#### Result Parameter

Parameter	Type	Description
Version	Date	Application Date

#### Examples

//Queries the application date

DEV:APPL:DATE?

2019-02-06

### DEvice:APPLication:VERSion?

Queries the application version number

#### Result Parameter

Parameter	Type	Description
Version	String	Application version number

#### Examples

//Queries the application version number

DEV:APPL:VERS?

"V1.3.1"

### DEvice:BIOS:VERSion?

Queries the BIOS version number

#### Result Parameter

Parameter	Type	Description
Version	String	Version number

#### Examples

//Queries the version number

DEV:BIOS:VERS?

"TQMxE38M.5.4.48.0028.15"

### DEvice:BOARD:CONTroller:BOOTloader:VERSion?

Queries the boardcontroller bootloader version

### Result Parameter

Parameter	Type	Description
Version	String	Version number

### Examples

//Queries the version number

DEV:BOAR:CONT:BOOT:VERS?

"V0.9.2"

## DEVice:BOARd:CONTroller:FIRMware:VERSion?

Queries the boardcontroller firmware version

### Result Parameter

Parameter	Type	Description
Version	String	Version number

### Examples

//Queries the version number

DEV:BOAR:CONT:FIRM:VERS?

"V1.0.2"

## DEVice:FPGA:BITStream:VERSion?

Queries the FPGA bitstream version

### Result Parameter

Parameter	Type	Description
Version	String	Version number

### Examples

//Queries the FPGA Bitstream version number

DEV:FPGA:BITS:VERS?

"528"

## DEVice:FPGA:PCIE:DRIVER:VERSion?

Queries the FPGA PCI Express driver version

### Result Parameter

Parameter	Type	Description
Version	String	Version number

### Examples

//Queries the version number

DEV:FPGA:PCIE:DRIV:VERS?

"1.8.0.0"

### DEVIce:OPERation:SYSTem:VERSion?

Queries the operation system version

#### Result Parameter

Parameter	Type	Description
Version	String	Version number

### Examples

//Queries the version number

DEV:OPER:SYST:VERS?

"V2.0.6"

### DEVIce:PART:NUMBer?

Queries the part number

#### Result Parameter

Parameter	Type	Description
Version	String	Part Number

### Examples

//Queries part number

DEV:PART:NUMB?

"3310/01"

### DEVIce:PRODuct:NAME?

Queries the product name

#### Result Parameter

Parameter	Type	Description
Version	String	Product name

### Examples

//Queries product name

DEV:PROD:NAME?

"SignalShark 3310"

### DEVIce:SERial:NUMBer?

Queries the serial number

### Result Parameter

Parameter	Type	Description
Version	String	Serial Number

### Examples

//Queries serial number

DEV:SER:NUMB?

"A-0054"

## Narda Command Reference - SETTings

Settings subsystem

### SETTings:ADJust:EQUalizer

Starts equalizer adjustment. This command has a timeout of 60s.

#### Examples

//Starts equalizer adjustment

SETT:ADJ:EQU

### SETTings:DATE?

Queries the date setting of the system

#### Result Parameter

Parameter	Type	Description
Date	Date	System Date

#### Examples

//Queries the system date

SETT:DATE?

2019-02-25

### SETTings:FAST:SCANs:ONLY

Allows fast scans only with limited BINS

#### Command Parameter

Parameter	Type	Description
Value	Bool	Fast scans only

#### Examples

//Disables fast scans for scan spectrum

SETT:FAST:SCAN:ONLY OFF

### SETTings:FAST:SCANs:ONLY?

Queries fast scan state

#### Result Parameter

Parameter	Type	Description
Value	Bool	Fast scans only



## Examples

//Queries fast scan state

SETT:FAST:SCAN:ONLY?

0

## SETTings:GNSS:SOURce

Sets the GNSS source

### Command Parameter

Parameter	Type	Description
Value	Enum	AUTO, INTERNAL, ANTENNA

## Examples

//Sets GNSS source to automatic mode

SETT:GNSS:SOUR AUTO

## SETTings:GNSS:SOURce?

Queries the GNSS source

### Result Parameter

Parameter	Type	Description
Value	Enum	AUTO, INTERNAL, ANTENNA

## Examples

//Queries GNSS source

SETT:GNSS:SOUR?

AUTO

## SETTings:MAGNetic:DECLination

Sets the magnetic declination

### Command Parameter

Parameter	Type	Description
Value	Double	0...180

## Examples

//Sets magnetic declination to 0

SETT:MAGN:DECL 0

## SETTings:MAGNetic:DECLination?

Queries the magnetic declination

### Result Parameter

Parameter	Type	Description
Value	Double	0...180

### Examples

//Queries magnetic declination

SETT:MAGN:DECL?

0

## SETTings:PPS:SOURce

Sets the PPS source

### Command Parameter

Parameter	Type	Description
Value	Enum	GNSS, EXTERNAL

### Examples

//Sets PPS source to active GNSS

SETT:PPS:SOUR GNSS

## SETTings:PPS:SOURce?

Queries the PPS source

### Result Parameter

Parameter	Type	Description
Value	Enum	GNSS, EXTERNAL

### Examples

//Queries PPS source

SETT:PPS:SOUR?

GNSS

## SETTings:REfERENCE:FREQuency:SOURce

Reference frequency source

### Command Parameter

Parameter	Type	Description
Value	Enum	INTERNAL, EXTERNAL

### Examples

//Sets reference frequency source to INTERNAL

SETT:REF:FREQ:SOUR INTERNAL

## SETTings:REFeRence:FREQuency:SOURce?

Queries the reference frequency source

### Result Parameter

Parameter	Type	Description
Value	Enum	INTERNAL, EXTERNAL

### Examples

//Queries the reference frequency source

SETT:REF:FREQ:SOUR?

INTERNAL

## SETTings:TIME?

Queries the system time?

### Result Parameter

Parameter	Type	Description
Time	Time	System Time

### Examples

//Queries the system time

SETT:TIME?

10:24:36

## SETTings:TIME:SYNC:NOW

Starts synchronization of systemtime with gnss time. This command has a timeout of 10s.

### Examples

//Starts system time synchronization

SETTings:TIME:SYNC:NOW

## SETTings:TSTamp:SYNC:MODE

Sets the timestamp synchronization mode

### Command Parameter

Parameter	Type	Description
Value	Enum	FREERUN, PPS

### Examples

//Sets timestamp synchronization to PPS

SETT:TST:SYNC:MODE PPS

### SETTings:TSTamp:SYNC:MODE?

Queries the timestamp synchronization mode

#### Result Parameter

Parameter	Type	Description
Value	Enum	FREERUN, PPS

#### Examples

//Queries timestamp synchronization mode

SETT:TST:SYNC:MODE?

PPS

### SETTings:TSTamp:SYNC:NOW

Starts timestamp synchronization and wait until sync finished. This command has a timeout of 10s.

#### Examples

//Starts timestamp synchronization

SETT:TST:SYNC:NOW

## Narda Command Reference - SENSE

Sense subsystem

### [SENSe:]ADC:OOR?

Queries ADC out of range state for the active measurement task

#### Result Parameter

Parameter	Type	Description
Value	Bool	ADC out of range state

#### Examples

```
//Queries ADC out of range state.
```

```
ADC:OOR?
```

```
0
```

### [SENSe:]ANTenna:AUTO:DATA?

Queries the data of an connected auto antenna

#### Result Parameter

Parameter	Type	Description
Manufacturer	String	Manufacturer of the antenna
DeviceName	String	Device name of the antenna
Model	String	Antenna model
SerialNumber	String	Serial number of the antenna
CalDate	Date	Calibration date of the antenna
MNOL	Float	MNOL
Channels	Enum	LPA, DDA, DLA1, DLA2, CH_1, CH_3,
NumberOfCalPoints	ULong	Number of calibration points
CalPoints	FlexArray[Double]	Calibration points in interleaved layout (Frequency in Hz   Attenuation in dB)

#### Examples

```
//Queries the data of an auto antenna
```

```
ANT:AUTO:DATA?
```

```
"Narda"," ", " 3502/01","PT-0013",2009-05-
```

```
26,16,3CH,19,800000000,49.7384262085,900000000,48.0928039551,1000000000,47.8465843201,1200000000,46.168006897,1400000000,44.5145111084,1600000000,43.0837478638,1800000000,41.3836212158,2000000000,41.7984733582,2200000000,41.7045021057,2450000000,42.9959602356,2700000000,43.2170906067,3000000000,44.9256896973,3500000000,45.9043731689,4000000000,45.6140975952,4500000000,46.843711853,5000000000,48.4728012085,5500000000,50.5361442566,5800000000,51.561214447,6000000000,51.7179794312
```

### [SENSe:]ANTenna:AUTO:INFO?

Queries the info data of an connected auto antenna

### Result Parameter

Parameter	Type	Description
ShortName	String	Short name of the antenna
LongName	String	Long name of the antenna
Fmin	String	Minimum frequency of the antenna
Fmax	String	Maximum frequency of the antenna
Property	Enum	E_FIELD, H_FIELD, CURRENT, E AND H_FIELD

### Examples

//Queries the info data of an auto antenna

ANT:AUTO:INFO?

"3AX 0.8-6G","Three-axis Antenna 800MHz - 6GHz","800 MHz","6 GHz",E\_FIELD

### [SENSe:]ANTenna:CHECksum?

Queries the checksum of the antenna connected to the SignalShark

### Result Parameter

Parameter	Type	Description
Value	ULong	Checksum of the connected antenna

### Examples

//Queries the checksum of the connected antenna.

ANT:CHEC?

2703025968

### [SENSe:]ANTenna:HANDle?

Queries the type of antenna handle connected to the SignalShark

### Result Parameter

Parameter	Type	Description
Value	Enum	NONE, HANDLE

### Examples

//Queries the antenna handle type

ANT:HAND?

NONE

### [SENSe:]ANTenna:ISOTropic:MANual

Enables/Disables the manual switching of antenna axis of isotropic antennas

## Command Parameter

Parameter	Type	Description
Enable	Bool	Manul switching of antenna axis enable state

## Examples

Enables the manual switching of antenna axis of an isotropic antenna

ANT:ISO:MAN ON

## [SENSe:]ANTenna:ISOTropic:MANual?

Queries the manual switching state of antenna axis of an isotropic antenna

## Result Parameter

Parameter	Type	Description
Enable	Bool	Manul switching of antenna axis enable state

## Examples

//Queries if the manual switching of antenna axis of an isotropic antenna is enabled

ANT:ISOT:MAN?

1

## [SENSe:]ANTenna:ISOTropic:MANual:AXIS

Sets the axis which is used for measurement in case that an isotropic antenna is used

## Command Parameter

Parameter	Type	Description
Axis	Enum	X_AXIS, Y_AXIS, Z_AXIS

## Examples

//Sets the axis which is used for measurement to the x axis

ANT:ISOT:MAN:AXIS X\_AXIS

## [SENSe:]ANTenna:ISOTropic:MANual:AXIS?

Queries which axis is used for measurement in case that an isotropic antenna is used

## Result Parameter

Parameter	Type	Description
Axis	Enum	X_AXIS, Y_AXIS, Z_AXIS, SINGLE, RSS

### Examples

//Queries the axis which is used for measurement

ANT:ISOT:MAN:AXIS?

X\_AXIS

### [SENSe:]ANTenna:MANual

Selects the manual antenna for the specified RF Input. An Antenna file must be edited with the built-in editor (Settings->Antennas->Create Antenna). The AntFile Parameter equals the “Long Name” in the editor

### Command Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4
AntFile	String	Name of the manual antenna to be loaded

### Examples

//Selects the manual antenna 'MyAntenna' for the RF Input 3

ANT:MAN 3, 'MyAntenna'

### [SENSe:]ANTenna:MANual?

Queries the manual antenna for the specified RF Input. An Antenna file must be edited with the built-in editor (Settings->Antennas->Create Antenna). The AntFile Parameter equals the “Long Name” in the editor

### Query Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4

### Result Parameter

Parameter	Type	Description
AntFile	String	Currently selected manual antenna

### Examples

//Queries the manual antenna for RF Input 2

ANT:MAN? 2

MyAntenna

### [SENSe:]ANTenna:MANual:LIST?

Queries the list of available manual antennas

### Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of manual antennas



## Examples

//Queries the list of available manual antennas

[SENSe:]ANTenna:MANual:LIST?

"No Antenna","MyAntenna\_L1"

## [SENSe:]ANTenna:POLarization?

Queries the polarization of the antenna connected to the antenna handle

### Result Parameter

Parameter	Type	Description
Value	Enum	NONE, HORIZONTAL, VERTICAL

## Examples

//Queries the polarization

ANT:POL?

NONE

## [SENSe:]ANTenna:TYPE?

Queries the type of antenna connected to the SignalShark

### Result Parameter

Parameter	Type	Description
Value	Enum	NONE, ADFA, DFA, ANTENNA, MANUAL

## Examples

//Queries the antenna type

ANT:TYPE?

NONE

## [SENSe:]ATTenuator

Attenuation for the RF input

### Command Parameter

Parameter	Type	Description
Value	Unit	Attenuation (dB)

## Examples

//Sets the RF attenuation to 5 dB

ATT 5

## [SENSe:]ATTenuator?

Queries the current attenuation

### Result Parameter

Parameter	Type	Description
Value	Unit	Attenuation in dB

### Examples

//Queries the current attenuation

ATT?

5

### [SENSe:]ATTenuator:LIST?

Queries a list of possible attenuator settings

### Result Parameter

Parameter	Type	Description
List	FlexArray[Float]	List of attenuator settings

### Examples

//Queries a list of attenuator settings

ATT:LIST?

31.5,31,30.5,30,29.5,29,28.5,28,27.5,27,26.5,26,25.5,25,24.5,24,23.5,23,22.5,22,21.5,21,20.5,20,19.5,19,18.5,18,17.5,17,16.5,16,15.5,15,14.5,14,13.5,13,12.5,12,11.5,11,10.5,10,9.5,9,8.5,8,7.5,7,6.5,6,5.5,5,4.5,4,3.5,3,2.5,2,1.5,1,0.5,0

### [SENSe:]CABLe:AUTO:DATA?

Queries the data of an connected auto cable

### Result Parameter

Parameter	Type	Description
Manufacturer	String	Manufacturer of the cable
DeviceName	String	Device name of the cable
Model	String	Cable model
SerialNumber	String	Serial number of the cable
CalDate	Date	Calibration date of the cable
NumberOfCalPoints	ULong	Number of calibration points
CalPoints	FlexArray[Double]	Calibration points in interleaved layout (Frequency in Hz   Attenuation in dB)
NumberGroupDelayCalPoints	ULong	Number of calibration points for group delay
GroupDelay	FlexArray[Double]	Group delay points in interleaved layout (Frequency in Hz   Delay in ns)

## Examples

//Queries the data of an auto antenna

CABLE:AUTO:DATA?

"Narda"," ", " 3602/02", " AC-0063",2016-07-

14,185,0,0,100000,0.0187890008092,200000,0.0356179997325,300000,0.0263260006905,500000,0.041590999  
8119,1000000,0.0675759986043,2000000,0.0919860005379,3000000,0.161724001169,5000000,0.1580030024  
05,10000000,0.208848997951,20000000,0.308171987534,40000000,0.446705013514,60000000,0.5540500283  
24,80000000,0.657486975193,100000000,0.717027008533,120000000,0.795826971531,140000000,0.8586689  
82983,160000000,0.929329991341,180000000,0.987419009209,200000000,1.04727995396,220000000,1.1202  
6000023,240000000,1.16901004314,260000000,1.22077000141,280000000,1.27967000008,300000000,1.3272  
3999023,320000000,1.3706099987,340000000,1.42557001114,360000000,1.47749996185, ...

## [SENSe:]CABLE:AUTO:INFO?

Queries the info data of an connected auto cable

### Result Parameter

Parameter	Type	Description
ShortName	String	Short name of the cable
LongName	String	Long name of the cable
Fmin	String	Minimum frequency of the cable
Fmax	String	Maximum frequency of the cable

## Examples

//Queries the info data of an auto cable

CABL:AUTO:INFO?

"SRM 5 m", " RF-Cable SRM - N 50 ohm - 5 m", "9 kHz", "6 GHz"

## [SENSe:]CABLE:CHECKsum?

Queries the checksum of the cable connected to the Signal Shark

### Result Parameter

Parameter	Type	Description
Value	ULong	Checksum of the connected cable

## Examples

//Queries the checksum of the connected cable.

CABL:CHEC?

2703025968

## [SENSe:]CABLE:MANual

Selects the manual cable for the specified RF Input. A Cable file must be edited with the built-in editor (Settings->Cables->New Cable). The CableFile Parameter equals the "Long Name" in the editor

### Command Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4
CableFile	String	Name of the manual cable to be loaded

### Examples

//Selects the manual cable 'MyCable' for the RF Input 3

CABL:MAN 3, 'MyCable'

### [SENSe:]CABLe:MANual?

Queries the manual cable for the specified RF Input. A Cable file must be edited with the built-in editor (Settings->Cables->New Cable). The CableFile Parameter equals the "Long Name" in the editor

### Query Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4

### Result Parameter

Parameter	Type	Description
CableFile	String	Currently selected manual cable

### Examples

//Queries the manual cable for RF Input 2

CABL:MAN? 2

MyCable

### [SENSe:]CABLe:MANual:LIST?

Queries the list of available manual cables

### Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of manual cables

### Examples

//Queries the list of available manual cables

CABL:MAN:LIST?

'No Cable','My Cable'

### [SENSe:]CABLe:TYPE?

Queries the type of cable connected to the SignalShark

## Result Parameter

Parameter	Type	Description
Value	Enum	NONE, CABLE, MANUAL

## Examples

//Queries the cable type

CABL:TYPE?

NONE

## [SENSe:]COMPass:DATA?

Compass values of the active compass

## Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

## Result Parameter

Parameter	Type	Description
Azimuth	Float	Azimuth in degree.
Elevation	Float	Elevation in degree.
Roll	Float	Roll in degree.

## Examples

//Queries a list of all compass values

COMP:DATA?

153.3,-3.6,178.3

## [SENSe:]EXTernal:DEvice

External device

## Command Parameter

Parameter	Type	Description
Value	Bool	External device state

## Examples

//Switches the external device ON

SENSe:EXTernal:DEvice ON

//Switches the external device OFF

EXT:DEV OFF

## [SENSe:]EXTernal:DEvice?

Queries the external device state

#### Result Parameter

Parameter	Type	Description
Value	Bool	External device state

#### Examples

//Queries the external device state

SENSe:EXTernal:DEvice?

0

#### [SENSe:]EXTernal:DEvice:MANual

Selects the manual external device for the specified RF Input. An External Device file must be edited with the built-in editor (Settings->Ext. Devices->New External Device). The ExtDevFile Parameter equals the "Long Name" in the editor

#### Command Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4
ExtDevFile	String	Name of the manual external device to be loaded

#### Examples

//Selects the manual external device 'MyExtDevice' for the RF Input 3

EXT:DEV:MAN 3, 'MyExtDevice'

#### [SENSe:]EXTernal:DEvice:MANual?

Queries the manual external device for the specified RF Input. An External Device file must be edited with the built-in editor (Settings->Ext. Devices->New External Device). The ExtDevFile Parameter equals the "Long Name" in the editor

#### Query Parameter

Parameter	Type	Description
RFInput	UShort	RF Input i.e. 1,2,3,4

#### Result Parameter

Parameter	Type	Description
ExtDevFile	String	Currently selected manual external device

#### Examples

//Queries the manual external device for RF Input 2

EXT:DEV:MAN? 2

MyExtDevice

#### [SENSe:]EXTernal:DEvice:MANual:LIST?

Queries the list of available manual external devices

## Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of manual external devices

## Examples

//Queries the list of available manual external devices

EXT:DEV:MAN:LIST?

'No External Device', 'MyExtDevice'

## [SENSe:]EXTeRnal:DEViCe:SWITChable?

Queries if the external device is switchable

## Result Parameter

Parameter	Type	Description
Value	Bool	External device switchable

## Examples

//Queries if the external device is switchable

SENSe:EXTeRnal:DEViCe:SWITChable?

1

//Queries if the external device is switchable

EXT:DEV:SWIT?

0

## [SENSe:]FREQuency:RANGe?

Queries the frequency range for active measurement task. The frequency range depends on the connected equipment.

## Result Parameter

Parameter	Type	Description
Fmin	Frequency	Minimum frequency in Hz
Fmax	Frequency	Maximum frequency in Hz

## Examples

//Queries frequency range in Hz

FREQ:RANG?

8000,8000000000

## [SENSe:]GNSS:DATA?

GNSS values of the active GNSS

## Command Reference Guide

### Command Descriptions

---

#### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

#### Result Parameter

Parameter	Type	Description
TimeOfLastTelegram	Time	UTC Time of last telegram including Milliseconds
DateOfLastTelegram	Date	UTC Date of last telegram
TimeOfLastPosFix	Time	UTC Time of last position fix including Milliseconds
DateOfLastPosFix	Date	UTC Date of last position fix
Quality	Enum	GPS, DGPS, GNSS, DGNSS, FROZEN, NOFIX
Fix	Enum	FIX2D, FIX3D, NOFIX
Satellites	UShort	Number of Satellites
Latitude	Double	Latitude
Longitude	Double	Longitude
Altitude	Float	Altitude in m
Speed	Float	Speed in m/s
Course	Float	Course in degree

#### Examples

//Queries a list of all GNSS values

GNSS:DATA?

13:22:30.900,2018-08-23,13:22:30.900,2018-08-23,GNSS,FIX3D,5,48.4584,9.231,445.9,0.082,0

#### [SENSe:]HOLD

Holds the measurement and initializes if measurement is stopped

#### Command Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

#### Examples

//Holds measurement with a timeout of 10 seconds for initializing

HOLD 10s

//Holds measurement with default timeout for initializing

SENSe:HOLD

#### [SENSe:]HOLD?

Holds the measurement and initializes if measurement is stopped



### Query Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

### Result Parameter

Parameter	Type	Description
ReturnCode	Long	Returns 0 if continuous run has started successfully
ReturnMsg	String	Returns 'No error' if successful

### Examples

//Holds measurement with a timeout of 10 seconds for initializing

HOLD? 10s

0,"No error"

//Holds measurement with default timeout for initializing

SENSe:HOLD?

0,"No error"

### [SENSe:]INPut

Selects the RF input

### Command Parameter

Parameter	Type	Description
Value	ULong	RF input (1,2,3,4)

### Examples

//Sets the RF input to input 1

INP 1

### [SENSe:]INPut?

Queries the RF input number

### Result Parameter

Parameter	Type	Description
Value	ULong	RF input (1,2,3,4)

### Examples

//Queries the RF input

INP?

1

### [SENSe:]PREamp

Preamplifier state of the antenna handle

#### Command Parameter

Parameter	Type	Description
Value	Bool	Preamplifier state

#### Examples

//Switches off the preamplifier

PRE OFF

### [SENSe:]PREamp?

Queries the state of the preamplifier of the antenna handle

#### Result Parameter

Parameter	Type	Description
Value	Bool	Preamplifier state

#### Examples

//Queries the current state of the preamplifier

PRE?

1

### [SENSe:]REFerence:LEVel

Reference level for the RF input

#### Command Parameter

Parameter	Type	Description
Value	Unit	Reference Level (dBm,...)

#### Examples

//Sets the RF reference level to -14

REF:LEV -14

### [SENSe:]REFerence:LEVel?

Queries the reference level for the RF input

#### Result Parameter

Parameter	Type	Description
Value	Unit	Reference Level

## Examples

//Queries the RF reference level

REF:LEV?

-14

## [SENSe:]REFerence:LEVel:LIST?

Queries a list of possible attenuator settings

### Result Parameter

Parameter	Type	Description
List	FlexArray[Float]	List of reference level settings

## Examples

//Queries a list of reference level settings

REF:LEV:LIST?

17.5,17,16.5,16,15.5,15,14.5,14,13.5,13,12.5,12,11.5,11,10.5,10,9.5,9,8.5,8,7.5,7,6.5,6,5.5,5,4.5,4,3.5,3,2.5,2,1.5,1,0.5,0,-0.5,-1,-1.5,-2,-2.5,-3,-3.5,-4,-4.5,-5,-5.5,-6,-6.5,-7,-7.5,-8,-8.5,-9,-9.5,-10,-10.5,-11,-11.5,-12,-12.5,-13,-13.5,-14

## [SENSe:]REFerence:LEVel:OFFSet

Sets the reference level offset. The flag [SENSe:]REFerence:LEVel:OFFSet:ENABle must be enabled for the setting of [SENSe:]REFerence:LEVel:OFFSet to take effect!

### Command Parameter

Parameter	Type	Description
Value	Unit	ReferenceLevelOffset (dB)

## Examples

//Sets the RF reference level offset to 5 dB

REF:LEV:OFFSet 5

## [SENSe:]REFerence:LEVel:OFFSet?

Queries the current reference level offset. The flag [SENSe:]REFerence:LEVel:OFFSet:ENABle must be enabled for the setting of [SENSe:]REFerence:LEVel:OFFSet to take effect!

### Result Parameter

Parameter	Type	Description
Value	Unit	ReferenceLevelOffset in dB

## Examples

//Queries the current reference level offset

REF:LEV:OFFSet?

5

### [SENSe:]REFerence:LEVel:OFFSet:ENABle

Enables/disables the reference level offset

#### Command Parameter

Parameter	Type	Description
Value	Bool	Reference level offset state

#### Examples

//Enables reference level offset

SENSe:REF:LEV:OFFS:ENAB ON

### [SENSe:]REFerence:LEVel:OFFSet:ENABle?

Queries the corresponding reference level offset state

#### Result Parameter

Parameter	Type	Description
Value	Bool	Reference level offset state

#### Examples

//Queries reference level offset state

SENSe:REF:LEV:OFFS:ENAB?

1

### [SENSe:]REFerence:LEVel:OFFSet:LIST?

Queries a list of possible reference level offset settings

#### Result Parameter

Parameter	Type	Description
List	FlexArray[Float]	List of reference level offset settings

#### Examples

//Queries a list of reference level offset settings

REF:LEV:OFFS:LIST?

40,39.99,39.98,39.97,39.96,39.95,39.94,39.93,39.92,39.91,39.9,39.89,39.88,39.87,39.86,39.85,39.84,39.83,39.82,39.81,39.8,39.79,39.78,39.77,39.76,39.75,39.74,39.73,39.72,39.71,39.7,39.69,39.68,39.67,39.66,39.65,39.64,39.63,39.62,39.61,39.6,39.59,39.58,39.57,39.56,39.55,39.54,39.53,39.52,39.51,39.5,39.49,39.48,39.47,39.46,39.45,39.44,39.43,39.42,...

### [SENSe:]RUN:CONTInuous

Starts a new continuous measurement

## Command Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

## Examples

//Starts a continuous measurement with a timeout of 10 seconds for initializing

RUN:CONT 10s

//Starts a continuous measurement with default timeout for initializing

SENSe:RUN:CONTinuous

## [SENSe:]RUN:CONTinuous?

Starts a new continuous measurement

## Query Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

## Result Parameter

Parameter	Type	Description
ReturnCode	Long	Returns 0 if continuous run has started successfully
ReturnMsg	String	Returns 'No error' if successful

## Examples

//Starts a continuous measurement with a timeout of 10 seconds for initializing

RUN:CONT? 10s

0,"No error"

//Starts a continuous measurement with default timeout for initializing

SENSe:RUN:CONTinuous?

0,"No error"

## [SENSe:]RUN:CONTinuous:RESet

Starts a new continuous measurement with reset

## Command Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

### Examples

//Starts a continuous measurement with reset and a timeout of 10 seconds for initializing

RUN:CONT:RES 10s

//Starts a continuous measurement with reset and default timeout for initializing

SENSe:RUN:CONTinuous:RESet

### [SENSe:]RUN:CONTinuous:RESet?

Starts a new continuous measurement with reset

### Query Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

### Result Parameter

Parameter	Type	Description
ReturnCode	Long	Returns 0 if continuous run has started successfully
ReturnMsg	String	Returns 'No error' if successful

### Examples

//Starts a continuous measurement with reset and a timeout of 10 seconds for initializing

RUN:CONT:RES? 10s

0,"No error"

//Starts a continuous measurement with reset and default timeout for initializing

SENSe:RUN:CONTinuous:RESet?

0,"No error"

### [SENSe:]RUN:SINGLE

Starts a new single measurement with reset

### Command Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing (s, ms)

### Examples

//Starts a single measurement with a timeout of 10 seconds for initializing

RUN:SING 10s

//Starts a single measurement with default timeout

SENSe:RUN:SINGLE

## [SENSe:]RUN:SINGle?

Starts a new single measurement with reset. The command processing ends when scan count equals scan number or the timeout value is reached.

### Query Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing and single measurement (s, ms)

### Result Parameter

Parameter	Type	Description
ReturnCode	Long	Returns 0 if single run has finished successfully
ReturnMsg	String	Returns 'No error' if successful

### Examples

//Starts a single measurement as a blocking command until initializing and measurement has finished with a timeout of 10 seconds.

RUN:SING? 10s

0,"No error"

//Starts a single measurement as a blocking command until initializing and measurement has finished with default timeout.

SENSe:RUN:SINGle?

0,"No error"

## [SENSe:]STOP

Stops the measurement

### Examples

//Stops the measurement

STOP

## [SENSe:]TSTamp:SYNC:DEViation?

Timestamp synchronization deviation

### Result Parameter

Parameter	Type	Description
Value	Double	Timestamp synchronisation deviation

### Examples

//Queries the timestamp synchronization deviation

SENse:TSTamp:SYNC:DEVIation?

5.12e-9

//Queries the timestamp synchronization deviation

TST:SYNC:DEV?

0

### [SENSe:]TSTamp:SYNC:FINE?

Timestamp fine synchronization state

#### Result Parameter

Parameter	Type	Description
Value	Bool	True if we have fine synchronisation

### Examples

//Queries the timestamp fine synchronization state

SENse:TSTamp:SYNC:FINE?

0

//Queries the timestamp fine synchronization state

TST:SYNC:FINE?

1

### [SENSe:]TSTamp:SYNC:STATE?

Timestamp synchronization state

#### Result Parameter

Parameter	Type	Description
Value	Enum	SYNCHRONIZED, UNSYNCHRONIZED, FREERUN

### Examples

//Queries timestamp synchronization state

TST:SYNC:STAT?

FREERUN



## Narda Command Reference - DISPlay

Display subsystem

### DISPlay:LEVelmeter:LMAX

Sets the maximum displayed level (x-axis) in the levelmeter view in actual unit

#### Command Parameter

Parameter	Type	Description
Value	Unit	LMAX (dBm, ...)

#### Examples

```
//Sets LMAX to -20 dBm
```

```
DISP:LEV:LMAX -20
```

### DISPlay:LEVelmeter:LMAX?

Queries the maximum displayed level (x-axis) in the levelmeter view in actual unit

#### Result Parameter

Parameter	Type	Description
Value	Unit	LMAX

#### Examples

```
//Queries LMAX
```

```
DISP:LEV:LMAX?
```

```
-20.00
```

### DISPlay:LEVelmeter:LRANge

Sets the maximum displayed level range (x-axis) in the levelmeter view in actual unit

#### Command Parameter

Parameter	Type	Description
Value	Unit	LRANge (dB)

#### Examples

```
//Sets LRANge to 100 dB
```

```
DISP:LEV:LRANge 100
```

### DISPlay:LEVelmeter:LRANge?

Queries the maximum displayed level range (x-axis) in the levelmeter view in actual unit

#### Result Parameter

Parameter	Type	Description
Value	Unit	LRANge

### Examples

//Queries LRANge?

DISP:LEV:LRAN?

100

## DISPlay:MAP:CENTer

Centers the map to the current GNSS position

### Examples

//Centers the map to the current GNSS position

DISPlay:MAP:CENTer

//Centers the map to the current GNSS position

DISP:MAP:CENT

## DISPlay:MAP:CENTer:POSition

Sets center point of the map

### Command Parameter

Parameter	Type	Description
Lat	Double	Latitude in degree (decimal)
Lon	Double	Longitude in degree (decimal)

### Examples

//Centers the map to the specified postion

DISPlay:MAP:CENT:POS 48.23196, 9.23589

## DISPlay:MAP:CENTer:POSition?

Queries the center point of the map

### Result Parameter

Parameter	Type	Description
Lat	Double	Latitude in degree (decimal)
Lon	Double	Longitude in degree (decimal)

### Examples

//Queries the center position of the map

DISPlay:MAP:CENTer:POSition?

48.23196, 9.23589

## DISPlay:MAP:LIST?

Queries a list of available maps

## Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of available maps

## Examples

//Queries a list of available maps

DISP:MAP:LIST?

'D: Default\Mapnik','D: Europa\Mapnik','Empty Map'

## DISPlay:MAP:SElect

Sets the selected map

## Command Parameter

Parameter	Type	Description
MapName	String	Name of the map to be selected

## Examples

//Sets the selected map

DISPlay:MAP:SElect 'Pfullingen'

## DISPlay:MAP:SElect?

Queries the name of the selected map

## Result Parameter

Parameter	Type	Description
MapName	String	Name of the currently selected map

## Examples

//Queries the selected map

DISP:MAP:SEL?

Default

## DISPlay:MAP:ZOOM:LEVel

Sets the zoom level

## Command Parameter

Parameter	Type	Description
ZoomLevel	UShort	zoom level

## Examples

//Sets the zoom level of the map to 13

DISP:MAP:ZOOM:LEV 13

### DISPlay:MAP:ZOOM:LEVel?

Queries the zoom level of the map

#### Result Parameter

Parameter	Type	Description
ZoomLevel	UShort	zoom level

#### Examples

//Queries the current zoom level of the map

DISP:MAP:ZOOM:LEV?

15

### DISPlay:PEAKtable:SHOW:TRANsmitter

Shows the transmitter table with peaktable

#### Command Parameter

Parameter	Type	Description
Value	Bool	Show transmitter table

#### Examples

//Shows transmitter table with peaktable

DISPlay:PEAKtable:SHOW:TRANsmitter true

### DISPlay:PEAKtable:SHOW:TRANsmitter?

Queries if the transmitter table is shown with peaktable

#### Result Parameter

Parameter	Type	Description
Value	Bool	Show transmitter table

#### Examples

//Queries if transmitter table is shown with peaktable

DISPlay:PEAKtable:SHOW:TRANsmitter?

1

### DISPlay:PEAKtable:SORT

Sort criteria for peaktable

#### Command Parameter

Parameter	Type	Description
Value	Enum	FREQUENCY, LEVEL

## Examples

//Sorts peaktable by LEVEL

DISP:PEAK:SORT LEVEL

## DISPlay:PEAKtable:SORT?

Queries the sort criteria for peaktable

### Result Parameter

Parameter	Type	Description
Value	Enum	FREQUENCY, LEVEL

## Examples

//Queries the sort criteria for peaktable

DISP:PEAK:SORT?

LEVEL

## DISPlay:PERStistence:LMAX

Sets the maximum displayed level (y-axis) in the persistence view in actual unit

### Command Parameter

Parameter	Type	Description
Value	Unit	LMAX (dBm, ...)

## Examples

//Sets LMAX to -20 dBm

DISP:PERS:LMAX -20

## DISPlay:PERStistence:LMAX?

Queries the maximum displayed level (y-axis) in the persistence view in actual unit

### Result Parameter

Parameter	Type	Description
Value	Unit	LMAX

## Examples

//Queries LMAX

DISP:PERS:LMAX?

-20.00

## DISPlay:PERStistence:LRANge

Sets the maximum displayed level range (y-axis) in the persistence view in actual unit

### Command Parameter

Parameter	Type	Description
Value	Unit	LRANge (dB)

### Examples

//Sets LRANge to 100 dB

DISP:PERS:LRAN 100

### DISPlay:PERsistence:LRANge?

Queries the maximum displayed level range (y-axis) in the persistence view in actual unit

### Result Parameter

Parameter	Type	Description
Value	Unit	LRANge

### Examples

//Queries LRANge

DISP:PERS:LRAN?

100

### DISPlay:SPECtrum:LMAX

Sets maximum displayed level (y-axis) in the spectrum view in actual unit

### Command Parameter

Parameter	Type	Description
Value	Unit	LMAX (dBm, ...)

### Examples

//Sets LMAX to -20 dBm

DISP:SPEC:LMAX -20

### DISPlay:SPECtrum:LMAX?

Queries the maximum displayed level (y-axis) in the spectrum view in actual unit

### Result Parameter

Parameter	Type	Description
Value	Unit	LMAX

### Examples

//Queries LMAX

DISP:SPEC:LMAX?

-20.00

## DISPlay:SPECTrum:LRANge

Sets the maximum displayed level range (y-axis) in the spectrum view in actual unit

### Command Parameter

Parameter	Type	Description
Value	Unit	LRANge (dB)

### Examples

//Sets LRANge to 100 dB

DISP:SPEC:LRAN 100

## DISPlay:SPECTrum:LRANge?

Queries the maximum displayed level range (y-axis) in the spectrum view in actual unit

### Result Parameter

Parameter	Type	Description
Value	Unit	LRANge

### Examples

//Queries LRANge

DISP:SPEC:LRAN?

100

## DISPlay:UNIT

Sets the actual level unit

### Command Parameter

Parameter	Type	Description
Value	Enum	<p>No antenna connected: dBm, dBV, dBmV, dBuV</p> <p>With antenna connected: dBm, dBV, dBmV, dBuV, W_m2, W_cm2, A_m, V_m, dBA_m, dBV_m, dBmV_m, dBuV_m</p>

### Examples

//Sets the unit to dBm

DISP:UNIT dBm

## DISPlay:UNIT?

Queries the actual level unit

### Result Parameter

Parameter	Type	Description
Value	Enum	No antenna connected: dBm, dBV, dBmV, dBuV  With antenna connected: dBm, dBV, dBmV, dBuV, W_m2, W_cm2, A_m, V_m, dBA_m, dBV_m, dBmV_m, dBuV_m

### Examples

//Queries the unit

DISP:UNIT?

dBm

## DISPlay:UNIT:LIST?

Queries a list of possible level units

### Result Parameter

Parameter	Type	Description
Data	FlexArray[Enum]	List of possible units

### Examples

//Queries a list of possible units

DISP:UNIT:LIST?

dBm,dBV,dBmV,dBuV



## Narda Command Reference - DLOGger

Datalogger subsystem

### DLOGger:SAVE:LOCalization

Datalogger save localization

#### Command Parameter

Parameter	Type	Description
FileName	String	Filename or empty string for auto naming
Comment	String	Comment for the file to save

#### Examples

//Saves localization to file with specific filename and comment

DLOG:SAVE:LOC 'MyFileName','This is a comment'

//Saves localization to file with no filename and no comment

DLOGger:SAVE:LOCalization "",

### DLOGger:WORKing:DIRectory

Datalogger get/set working directory

#### Command Parameter

Parameter	Type	Description
Path	String	Path to working directory

#### Examples

//Sets the working directory

DLOG:WORK:DIR 'D:\\Narda\_SignalShark\\Datalogger\\Measurement Data'

### DLOGger:WORKing:DIRectory?

Datalogger get/set working directory

#### Result Parameter

Parameter	Type	Description
Path	String	Path to working directory

#### Examples

//Queries the working directory

DLOGger:WORKing:DIRectory?

'D:\\Narda\_SignalShark\\Datalogger\\Measurement Data'

### DLOGger:CONFig:TASK:SAVE

Datalogger configuration

### Command Parameter

Parameter	Type	Description
Value	Enum	ALL, SELECTED

### Examples

//Configures datalogger to save selected task only

DLOG:CONF:TASK:SAVE SELECTED

### DLOGger:CONFig:TASK:SAVE?

Queries the datalogger configuration

### Result Parameter

Parameter	Type	Description
Value	Enum	ALL, SELECTED

### Examples

//Queries the datalogger save configuration

DLOG:CONF:TASK:SAVE?

SELECTED

### DLOGger:DIRectory:ADD?

Creates a new directory and returns the directory path

### Query Parameter

Parameter	Type	Description
Value	String	Path to the new directory

### Result Parameter

Parameter	Type	Description
Value	String	Path to the new directory

### Examples

//Creates a new directory and returns the directory path

DLOGger:DIRectory:ADD? 'E:/Narda\_SignalShark/Datalogger/Data/Bearing'

E:/Narda\_SignalShark/Datalogger/Data/Bearing\_1

### DLOGger:DIRectory:DElete

Deletes the given directory

### Command Parameter

Parameter	Type	Description
Directory	String	Directory to be deleted

## Examples

//Deletes the given directory

DLOGger:DIRectory:DElete 'D:/Narda\_SignalShark/Datalogger/Data/Bearing1'

## DLOGger:DIRectory:DElete:ALL

Deletes all files in the given directory

### Command Parameter

Parameter	Type	Description
Directory	String	Directory of which the content will be deleted

## Examples

//Deletes all files in the given directory

DLOGger:DIRectory:DElete:ALL 'E:/Narda\_SignalShark/Datalogger/Data/Bearing'

## DLOGger:DIRectory:LIST?

Queries a list of all directories within the data logger

### Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of directories

## Examples

//Queries a list of all directories within the data logger

DLOGger:DIRectory:LIST?

## DLOGger:RECall:DATaset

Recalls a dataset. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset to take effect!

WARNING, this command is only recommended to be used in combination with other Querie commands. Invalid combination of commands could lead to a crash of the SignalShark application!

### Command Parameter

Parameter	Type	Description
DataSet	String	The name of the dataset

## Examples

//Recalls the dataset with the name 'dataset'

DLOG:REC:DAT 'dataset'

## DLOGger:RECall:DATaset?

Returns the name of the current recalled dataset. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset to take effect!

#### Result Parameter

Parameter	Type	Description
DataSet	String	The name of the recalled dataset

#### Examples

//Returns the name of the current recalled dataset

DLOG:REC:DAT?

dataset

#### DLOGger:RECall:DATaset:CLOSe

Closes the current dataset recall. The flag DLOGger:RECall:ENABled must be enabled for the setting of DLOGger:RECall:DATaset:CLOSe to take effect!

#### Examples

//Closes the current dataset recall

DLOGger:RECall:DATaset:CLOSe

#### DLOGger:RECall:ENABled

Enables/Disables recalls in remote.

WARNING, this command is only recommended to be used in combination with other Querie commands. Invalid combination of commands could lead to a crash of the SignalShark application!

#### Command Parameter

Parameter	Type	Description
Enabled	Bool	Recall enabled

#### Examples

//Enables recalls in remote

DLOG:REC:ENAB ON

#### DLOGger:RECall:ENABled?

Queries the recall enabled state

#### Result Parameter

Parameter	Type	Description
Enabled	Bool	Recall enabled

#### Examples

//Queries the recall enabled state

DLOG:REC:ENAB?

1

## DLOGger:SAVE:DATaset

Datalogger save dataset

### Command Parameter

Parameter	Type	Description
FileName	String	Filename or empty string for auto naming
Comment	String	Comment for the file to save

### Examples

//Saves dataset to file with specific filename and comment

DLOG:SAVE:DAT 'MyFileName','This is a comment'

//Saves dataset to file with no filename and no comment

DLOGger:SAVE:DATaset "",

## DLOGger:SAVE:EXternal:BEARing

Datalogger save external bearing

### Command Parameter

Parameter	Type	Description
Ftune	Double	Frequency
Latitude	Double	GNSS Latitude
Longitude	Double	GNSS Longitude
Azimuth	Double	Azimuth in degree: 0°...360°
OptionalParameters	Optional[String,String,String,Date,Time,Double,Double,Double]	Filename or empty string for auto naming, Optional Comment or empty string, Source or empty string, Date of the external bearing, Time of the external bearing, Elevation in degree: -90°...90°, DF Quality in percent 0%...100%, RMS level value in dBm

### Examples

//Saves external bearing to file

DLOG:SAVE:EXT:BEAR 300.0e6, 48.472, 9.2212, 300.09, 'MyBearing', 'Some comment', 'Some Source', 21.03.2015, 14:07:09, 0.3, 100.0,-98.4

//Saves external bearing to file

DLOGger:SAVE:EXternal:BEARing 51.2e6,48.472,9.220,100.0

## DLOGger:SAVE:SCReenshot

Saves screenshot to file

## Command Reference Guide

### Command Descriptions

---

#### Command Parameter

Parameter	Type	Description
FileName	String	Filename or empty string for auto naming
Comment	String	Comment for the file to save

#### Examples

//Saves screenshot to file with specific filename and comment

DLOG:SAVE:SCR 'MyFileName','This is a comment'

## Narda Command Reference - SYSTem

System subsystem

### SYSTem:AUDio:MUTE

Sets the system audio mute setting

#### Command Parameter

Parameter	Type	Description
Value	Bool	Audio mute

#### Examples

```
//Enables audio mute
```

```
SYST:AUD:MUTE ON
```

### SYSTem:AUDio:MUTE?

Queries the system audio mute setting

#### Result Parameter

Parameter	Type	Description
Value	Bool	Audio mute

#### Examples

```
//Queries audio mute
```

```
SYST:AUD:MUTE?
```

```
1
```

### SYSTem:AUDio:VOLume

Sets the system audio volume setting

#### Command Parameter

Parameter	Type	Description
Value	Double	Audio volume 0%...100%

#### Examples

```
//Sets the audio volume to 90%
```

```
SYST:AUD:VOL 90
```

### SYSTem:AUDio:VOLume?

Queries the system audio volume setting

#### Result Parameter

Parameter	Type	Description
Value	Double	Audio volume

### Examples

//Queries the audio volume

SYST:AUD:VOL?

90

## SYSTem:COMManD:FILTer?

Queries the filter flags of command processing

### Result Parameter

Parameter	Type	Description
Value	ULong	Filter Flags

### Examples

//Queries the command filter flags

SYST:COMM:FILT?

0

## SYSTem:COMManD:PROGress?

Queries the progress of command processing

### Result Parameter

Parameter	Type	Description
StepNumber	Double	Progress step number
StepCount	Double	Progress step count
ProgressInfo	String	Progress info

### Examples

//Queries command progress

SYST:COMM:PROG?

0,0,""

## SYSTem:MEMory:STATus?

Queries the memory status of the system

### Result Parameter

Parameter	Type	Description
MemoryLoad	ULong	RAM load in percent
TotalPhysical	ULong	Total physical RAM in bytes
AvailablePhysical	ULong	Total physical RAM available in bytes
UsedByProcess	ULong	RAM used by application in bytes



## Examples

//Queries the memory status of the system

SYSTem:MEMory:STATus?

24,4181934080,3157458944,418177024

## SYSTem:REMOte:COMManD:LIST?

Queries a list with all available commands in the narda remote parser

### Query Parameter

Parameter	Type	Description
Optional	Optional[String]	e.g. 'ShowParameters'

### Result Parameter

Parameter	Type	Description
Value	String	List of available scpi commands

## Examples

//Queries a list of available commands with parameter description

SYST:REM:COMM:LIST? 'ShowParameters'

...BEARing:DFSQuelch:ENABLE Bool;...

//Queries a list of available commands

SYSTem:REMOte:COMManD:LIST?

\*IDN?;\*OPC?;\*RST;\*WAI;ABORT;SYSTem:ERRor:CODE[:NEXT]?;...

## SYSTem:REMOte:DISPlay

Enables or disables GUI views while remote on

### Command Parameter

Parameter	Type	Description
Value	Bool	Remote Display

## Examples

//Disables remote display

SYST:REM:DISP 0

## SYSTem:REMOte:DISPlay?

Queries the remote display state

### Result Parameter

Parameter	Type	Description
Value	Bool	Remote Display

### Examples

//Queries the remote display value

SYST:REM:DISP?

0

## SYSTem:REMOte:LOG

Enables or disables remote logfile

### Command Parameter

Parameter	Type	Description
Value	Bool	Remote Logfile

### Examples

//Disables remote logfile

SYST:REM:LOG OFF

## SYSTem:REMOte:LOG?

Queries the remote logfile state

### Result Parameter

Parameter	Type	Description
Value	Bool	Remote Logfile

### Examples

//Queries the remote logfile state

SYST:REM:LOG?

0

## SYSTem:REMOte:LOG:CONFig

Sets the remote logfile configuration

### Command Parameter

Parameter	Type	Description
Receive	Bool	True for showing receive data
Transmit	Bool	True for showing transmit data
Timestamp	Bool	True for showing timestamp info
Connection	Bool	True for showing connection info
LineSize	ULong	Maximum line size, 0 = disable line size limit
LineCount	ULong	Maximum line count, 0 = disable line count limit

## Examples

//Sets remote logfile configuration

SYST:REM:LOG:CONF true, true, true, true, 0, 0

## SYSTem:REMOte:LOG:CONFig?

Queries the remote logfile configuration

### Result Parameter

Parameter	Type	Description
Receive	Bool	True for showing receive data
Transmit	Bool	True for showing transmit data
Timestamp	Bool	True for showing timestamp info
Connection	Bool	True for showing connection info
LineSize	ULong	Maximum line size, 0 = disable line size limit
LineCount	ULong	Maximum line count, 0 = disable line count limit

## Examples

//Queries remote logfile configuration

SYST:REM:LOG:CONF?

1,1,1,1,0,0

## SYSTem:REMOte:SLEEp?

Lets parser sleep for a specific timespan and returns true if successful.

### Query Parameter

Parameter	Type	Description
Time	Timespan	Timespan for sleeping

### Result Parameter

Parameter	Type	Description
ReturnCode	Bool	Returns 1 if succesfull

## Examples

//Lets the parser sleep for 100 milliseconds

SYSTem:REMOte:SLEEp? 100ms

1

//Lets the parser sleep for 1 second

SYST:REM:SLE? 1s

1

### SYSTem:REMOte:TIMEout

SCPI default remote timeout (default factory setting = 3 seconds)

#### Command Parameter

Parameter	Type	Description
Timeout	Timespan	Remote timeout (s, ms)

#### Examples

//Sets remote timeout to 10 seconds

SYST:REM:TIM 10s

### SYSTem:REMOte:TIMEout?

Queries the SCPI default remote timeout

#### Result Parameter

Parameter	Type	Description
Timeout	Timespan	Remote timeout in s

#### Examples

//Queries the remote timeout

SYST:REM:TIM?

10

### SYSTem:REMOte:TIMEout:RESet

Resets remote timeout of all commands to the default value

#### Examples

//Resets remote timeout of all commands to the default value

SYST:REM:TIM:RES

### SYSTem:SHUTdown

Shuts down or restarts the instrument

#### Command Parameter

Parameter	Type	Description
Value	Enum	Shutdown Mode (SYSTEM_RESTART, SYSTEM_LOG_OFF, APP_QUIT, APP_RESTART)

#### Examples

//Restarts the instrument

SYSTem:SHUTdown SYSTEM\_RESTART

## SYSTem:STARtup:DATE?

Startup Date of the system

### Result Parameter

Parameter	Type	Description
Date	Date	System startup Date

### Examples

//Queries the system startup date

SYSTem:STARtup:DATE?

2019-07-26

//Queries the system startup date

SYST:STAR:DATE?

2019-07-26

## SYSTem:STARtup:SEConds?

Seconds since last start of the system

### Result Parameter

Parameter	Type	Description
Seconds	ULongLong	Seconds since last start of the system

### Examples

//Queries the time since last start of the system in seconds

SYSTem:STARtup:SEConds?

265050

//Queries the time since last start of the system in seconds

SYST:STAR:SEC?

265050

## SYSTem:STARtup:TIME?

Startup Time of the system

### Result Parameter

Parameter	Type	Description
Time	Time	System startup Time

#### Examples

//Queries the system startup time

SYSTem:STARtup:TIME?

09:22:25

//Queries the system startup time

SYST:STAR:TIME?

09:22:25

## Narda Command Reference - NETWork

Network settings subsystem

### NETWork:MAC:ADDRess?

Queries the Network MAC address

#### Result Parameter

Parameter	Type	Description
Address	String	Network MAC address

#### Examples

//Queries the Network MAC address

NETW:MAC:ADDR?

"00-D0-93-2C-93-78"

## Narda Command Reference - TASK

Task subsystem

### TASK:ADD?

Adds a new measurement task to the instrument

#### Query Parameter

Parameter	Type	Description
Type	String	Task type e.g. 'SPECTRUM', 'RT_SPECTRUM', 'RT_STREAMING', 'AUTO_DF'

#### Result Parameter

Parameter	Type	Description
Name	String	Name of the new task

#### Examples

//Adds a new realtime spectrum task and queries the name of the task

TASK:ADD? 'RT\_SPECTRUM'

"RT Spectrum"

### TASK:DELeTe

Deletes a measurement task

#### Command Parameter

Parameter	Type	Description
Name	String	Task name to delete

#### Examples

//Deletes the measurement task with the name 'Spectrum01'

TASK:DEL 'Spectrum01'

### TASK:DELeTe:ALL

Deletes all a measurement tasks

#### Examples

//Deletes all measurement tasks

TASK:DEL:ALL

### TASK:LIST?

Queries a list of measurement tasks and task names



## Result Parameter

Parameter	Type	Description
List	FlexArray[String,String]	List of all measurement tasks (Type, Name)

## Examples

//Queries a list of all measurement tasks

TASK:LIST?

"RT\_SPECTRUM","RT Spectrum","RT\_STREAMING","RT Streaming","SPECTRUM","Spectrum","SPECTRUM","Spectrum01"

## TASK:MOVE

Moves a measurement task to another position

## Command Parameter

Parameter	Type	Description
Name	String	Task to move
NameNewPos	String	Destination task to move in front of, or empty string to move to the end

## Examples

//Moves the measurement task 'Spectrum01' to the end

TASK:MOVE 'Spectrum01',"

## TASK:NEW?

Adds a new measurement task and deletes existing tasks. This command has a timeout of 10s

## Query Parameter

Parameter	Type	Description
Type	String	Task type e.g. 'SPECTRUM', 'RT_SPECTRUM', 'RT_STREAMING', 'AUTO_DF'

## Result Parameter

Parameter	Type	Description
Name	String	Name of the new task

## Examples

//Adds a new realtime spectrum task and queries the name of the task

TASK:NEW? 'RT\_SPECTRUM'

"RT Spectrum"

## TASK:REName

Renames a measurement task

### Command Parameter

Parameter	Type	Description
Name	String	Task to rename
newName	String	New task name

### Examples

//Renames the measurement task 'Spectrum01' with the name 'mySpectrum01'

TASK:REN 'Spectrum01', 'mySpectrum01'

### TASK:REPLace?

Replaces an existing measurement task

### Query Parameter

Parameter	Type	Description
Name	String	Name of task to replace
Type	String	Task type of new task e.g. 'RT_SPECTRUM', 'AUTO_DF'

### Result Parameter

Parameter	Type	Description
Name	String	Name of the new task

### Examples

//Replaces the measurement task with the name 'Spectrum01' with a new task of type 'RT\_SPECTRUM'

TASK:REPL? 'Spectrum01','RT\_SPECTRUM'

"RT Spectrum"

### TASK:SELection

Selects a measurement task

### Command Parameter

Parameter	Type	Description
Name	String	Task name to select

### Examples

//Selects the measurement task with the name 'Spectrum01'

TASK:SEL 'Spectrum01'

### TASK:SELection?

Queries the selected measurement task

### Result Parameter

Parameter	Type	Description
Type	String	Task type
Name	String	Task name

### Examples

//Queries the selected measurement task

TASK:SEL?

"SPECTRUM","Spectrum01"

### TASK:STATE?

Queries the task state of the selected task

### Result Parameter

Parameter	Type	Description
Value	Enum	Task state (RUN, HOLD, STOP)

### Examples

//Queries the task state of the selected task

TASK:STAT?

RUN

## Narda Command Reference - VIEW

View subsystem

### VIEW:ADD?

Adds a new view to the active measurement task

#### Query Parameter

Parameter	Type	Description
Reference	UShort	Index of the reference view to insert the new view next to
Direction	Enum	Position of the new view relative to the reference view (LEFT,RIGHT,ABOVE,BELOW)
Type	String	Type of view e.g. 'SPECTRUM', 'PEAK_TABLE', 'LEVEL', 'PERSISTENCE', 'SPECTROGRAM', 'BEARING', 'MAP'

#### Result Parameter

Parameter	Type	Description
NewView	UShort	Index of the new view added

#### Examples

//Adds a new peaktable view to the left of the view with index 1

VIEW:ADD? 1,LEFT,'PEAK\_TABLE'

2

### VIEW:DELeTe

Deletes a view from the active measurement task

#### Command Parameter

Parameter	Type	Description
View	UShort	Index of the view to delete

#### Examples

//Deletes the view with index 1

VIEW:DEL 1

### VIEW:LIST?

Queries a list of views of the active measurement task

#### Result Parameter

Parameter	Type	Description
List	FlexArray[String,UShort]	List of all views (Type, Index)

[9 Command Groups ^](#)

## Examples

//Queries a list of views of the active measurement task

VIEW:LIST?

"PEAK\_TABLE",2,"SPECTRUM",1

## VIEW:REPLace

Replaces a view of the active measurement task

### Command Parameter

Parameter	Type	Description
View	UShort	Index of the view to replace
Type	String	Type of view e.g. 'PEAK_TABLE'

## Examples

//Replaces the view with index 1 with a peaktbale view

VIEW:REPL 1,'PEAK\_TABLE'

## VIEW:SElect

Selects a view from the active measurement task

### Command Parameter

Parameter	Type	Description
View	UShort	Index of the view to select

## Examples

//Selects the view with index 1

VIEW:SEL 1

## VIEW:SElect?

Queries the index of the selected view of the active measurement task

### Result Parameter

Parameter	Type	Description
View	UShort	Index of the selected view

## Examples

//Queries the index of the selected view of the active measurement task

VIEW:SEL?

1

## VIEW:SIZE:MAXimize

Maximizes the selected view

### Examples

//Maximizes the selected view

VIEW:SIZE:MAX

### VIEW:SIZE:RESTore

Restores the size of the selected view

### Examples

//Restores the size of the selected view

VIEW:SIZE:REST

## Narda Command Reference - SPECTrum

Spectrum subsystem

### SPECTrum:DATA:ALL?

Queries the spectrum configuration and values for multiple traces. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)

#### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
ConfigID	Enum	Optional: CONFIG
ConfigElements	ULong	Number of ConfigID Elements
ScanSteps	ULong	Number of Scansteps
ScanTime	Double	Average Scan Time / Meas Time
TotalBins	ULong	Number of Bins
FreqStart	Frequency	Start frequency of the X-Axis
FreqStep	Frequency	Frequency step of the X-Axis
TraceID	Enum	Optional: RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS
TraceElements	ULong	Number of TraceID Elements
Overdriven	Bool	Overdriven flag
NotRealtime	Bool	NotRealtime flag
LevelValues	FlexArray[Float]	List of power levels in actual unit

#### Examples

//Queries all activated traces of the spectrum

SPECTrum:DATA:ALL?

0,1551350795,722909033,3306,CONFIG,5,1,0.01,51,90000000,400000,RMS,53,0,0,-66.20,-67.63,-73.90,-79.52,-71.85,-65.89,-65.97,-72.14,-81.58,-81.68,-75.74,-68.35,-66.88,-71.39,-82.09,-87.16,-86.25,-73.73,-62.21,-57.57,-58.99,-66.72,-82.28,-92.28,-92.18,-91.34,-85.38,-76.03,-71.63,-72.93,-76.87,-76.62,-74.96,-74.30,-78.34,-69.98,-60.43,-57.41,-60.11,-65.24,-68.36,-75.68,-86.24,-84.01,-79.83,-81.25,-88.08,-93.73,-93.82,-93.17,-92.43

### SPECTrum:DATA:COUNT?

Queries the spectrum data count (number of bins)

### Result Parameter

Parameter	Type	Description
Value	ULong	Number of bins

### Examples

//Queries the number of bins for the spectrum results

SPEC:DATA:COUN?

27

## SPECtrum:DATA:FREQuency:START?

Queries spectrum data start frequency

### Result Parameter

Parameter	Type	Description
Value	Frequency	Start frequency of the X-Axis

### Examples

//Queries the start frequency for the spectrum results

SPEC:DATA:FREQ:STAR?

87400000

## SPECtrum:DATA:FREQuency:STEP?

Queries spectrum data frequency step

### Result Parameter

Parameter	Type	Description
Value	Frequency	Frequency step of the X-Axis

### Examples

//Queries the frequency step for the spectrum results

SPEC:DATA:FREQ:STEP?

200000

## SPECtrum:DATA:LEVel?

Queries spectrum level data

### Query Parameter

Parameter	Type	Description
Name	Enum	RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS



## Result Parameter

Parameter	Type	Description
Data	FlexArray[Float]	List of power levels in actual unit

## Examples

//Queries the spectrum data results for AvR

SPEC:DATA:LEV? AvR

-92.93,-84.10,-78.49,-78.52,-84.42,-93.50,-92.02,-89.82,-83.25,-70.33,-64.19,-63.79,-61.04,-55.48,-55.48,-61.55,-69.25,-67.60,-70.60,-80.08,-93.79,-87.75,-70.16,-60.62,-57.62,-60.60,-70.13

## SPECTrum:DATA:OVERdriven?

Queries the spectrum data overdriven flag

## Result Parameter

Parameter	Type	Description
Value	Bool	Overdriven flag

## Examples

//Queries the overdriven flag

SPEC:DATA:OVER?

0

## SPECTrum:DATA:REALtime?

Queries the spectrum data realtime flag for traces: MnR, AvR, MxR, MxP, MnP, MxA, MxS

## Result Parameter

Parameter	Type	Description
Value	Bool	Realtime flag

## Examples

//Queries the realtime flag

SPEC:DATA:REAL?

1

## SPECTrum:DATA:UPDate?

Update is available in RUN and HOLD mode (see DATA:UPDate Commands)

## Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong, Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

## Command Reference Guide

### Command Descriptions

---

#### Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.

#### Examples

//Spectrum data update with old scancount = 10 and timeout = 100ms

SPEC:DATA:UPD? 10,100ms

25244

//Forces Spectrum data update because we do not have old scannumber

SPECtrum:DATA:UPDate?

24548

//Spectrum data update with old scannumber = 0

SPECtrum:DATA:UPDate? 0

24792

//Forces Spectrum data update with timeout = 100ms

SPECtrum:DATA:UPDate? NaN,100ms

25024

## SPECtrum:FREQuency:CENTer

Sets the center frequency for spectrum

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Center frequency (Hz, kHz, MHz, GHz)

#### Examples

//Sets the center frequency in MHz

SPEC:FREQ:CENT 51.2 MHz

## SPECtrum:FREQuency:CENTer?

Queries the center frequency

#### Result Parameter

Parameter	Type	Description
Value	Frequency	Center frequency in Hz

#### Examples

//Queries the center frequency

SPEC:FREQ:CENT?

51200000

## SPECTrum:FREQuency:CENTer:STEP

Sets the center frequency step for spectrum

### Command Parameter

Parameter	Type	Description
Value	Frequency	Center frequency step (Hz, kHz, MHz, GHz)

### Examples

//Sets the center frequency step in MHz

SPEC:FREQ:CENT:STEP 1 MHz

## SPECTrum:FREQuency:CENTer:STEP?

Queries the center frequency step for spectrum

### Result Parameter

Parameter	Type	Description
Value	Frequency	Center frequency step in Hz

### Examples

//Queries the center frequency step size

SPEC:FREQ:CENT:STEP?

1000000

## SPECTrum:FREQuency:CENTer:TUNE:COUPling

Automatic coupling of Fcent and Ftune

### Command Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Sets the Fcent Ftune coupling to OFF

SPEC:FREQ:CENT:TUNE:COUP OFF

## SPECTrum:FREQuency:CENTer:TUNE:COUPling?

Queries the coupling state of Fcent and Ftune

### Result Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Queries the Fcent Ftune coupling state

SPEC:FREQ:CENT:TUNE:COUP?

0

## SPECTrum:FREQuency:ENTRy:MODE

Sets the entry mode for Fcent, Fspan or Fstart, Fstop

### Command Parameter

Parameter	Type	Description
Value	Enum	Entry mode (FSTART_FSTOP,FCENT_FSPAN)

### Examples

//Sets entry mode to FCENT\_FSPAN

SPEC:FREQ:ENTR:MODE FCENT\_FSPAN

## SPECTrum:FREQuency:ENTRy:MODE?

Queries the entry mode for Fcent, Fspan or Fstart, Fstop

### Result Parameter

Parameter	Type	Description
Value	Enum	Entry mode (FSTART_FSTOP,FCENT_FSPAN)

### Examples

//Queries the entry mode

SPEC:FREQ:ENTR:MODE?

FCENT\_FSPAN

## SPECTrum:FREQuency:SPAN

Frequency span for spectrum

### Command Parameter

Parameter	Type	Description
Value	Frequency	Frequency span (Hz, kHz, MHz, GHz)

### Examples

//Sets the frequency span in MHz

SPEC:FREQ:SPAN 40 MHz

## SPECTrum:FREQuency:SPAN?

Queries the frequency span

### Result Parameter

Parameter	Type	Description
Value	Frequency	Frequency span in Hz

### Examples

//Queries the frequency span

SPEC:FREQ:SPAN?

40000000

## SPECTrum:FREQuency:STARt

Sets the Start frequency for spectrum

### Command Parameter

Parameter	Type	Description
Value	Frequency	Start frequency (Hz, kHz, MHz, GHz)

### Examples

//Sets the start frequency in GHz

SPEC:FREQ:STAR 1 GHz

## SPECTrum:FREQuency:STARt?

Queries the start frequency, in Hz

### Result Parameter

Parameter	Type	Description
Value	Frequency	Start frequency in Hz

### Examples

//Queries the start frequency in Hz

SPEC:FREQ:STAR?

1000000000

## SPECTrum:FREQuency:STOP

Sets the Stop frequency for spectrum

### Command Parameter

Parameter	Type	Description
Value	Frequency	Stop frequency (Hz, kHz, MHz, GHz)

### Examples

//Sets the stop frequency in Hz

SPEC:FREQ:STOP 2e9

## SPECTrum:FREQuency:STOP?

Queries the stop frequency, in Hz

### Result Parameter

Parameter	Type	Description
Value	Frequency	Stop frequency in Hz

### Examples

//Queries the stop frequency in Hz

SPEC:FREQ:STOP?

2000000000

## SPECTrum:MEASurement:TIME

Measurement time for spectrum

### Command Parameter

Parameter	Type	Description
Value	Timespan	Measurement time (h, m, s, ms)

### Examples

//Sets the measurement time to 10ms

SPEC:MEAS:TIME 10 ms

## SPECTrum:MEASurement:TIME?

Queries the measurement time

### Result Parameter

Parameter	Type	Description
Value	Timespan	Measurement time in s

### Examples

//Queries the measurement time

SPEC:MEAS:TIME?

0.01

## SPECTrum:MEASurement:TIME:ENTRy:MODE

Sets the measurement time entry mode, absolute or relative

### Command Parameter

Parameter	Type	Description
Value	Enum	Measurement time entry mode (Absolute, Relative)

### Examples

//Sets the measurement time entry mode to ABSOLUTE

SPECTrum:MEASurement:TIME:ENTRy:MODE Absolute

## SPECTrum:MEASurement:TIME:ENTRy:MODE?

Queries the measurement time entry mode, absolute or relative

### Result Parameter

Parameter	Type	Description
Value	Enum	Measurement time entry mode (Absolute, Relative)

### Examples

//Queries the measurement time entry mode

SPEC:MEAS:TIME:ENTR:MODE?

ABSOLUTE

## SPECTrum:RBW

Sets the resolution bandwidth for spectrum

### Command Parameter

Parameter	Type	Description
Value	Frequency	Resolution Bandwidth (Hz, kHz, MHz, GHz)

### Examples

//Sets the bandwidth in kHz

SPEC:RBW 100 kHz

## SPECTrum:RBW?

Queries the bandwidth, in Hz

### Result Parameter

Parameter	Type	Description
Value	Frequency	Resolution Bandwidth in Hz

### Examples

//Queries the bandwidth in Hz

SPEC:RBW?

100000

## SPECTrum:RBW:AUTO

Bandwidth automatic coupling to the span

### Command Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Sets the bandwidth automatic coupling to OFF

SPEC:RBW:AUTO OFF

## SPECTrum:RBW:AUTO?

Queries the bandwidth automatic coupling state

### Result Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Queries the bandwidth automatic coupling state

SPEC:RBW:AUTO?

1

## SPECTrum:RBW:ENTRy:MODE

Sets the bandwidth entry mode; absolute or relative

### Command Parameter

Parameter	Type	Description
Value	Enum	RBW entry (Absolute, Relative)

### Examples

//Sets the bandwidth entry to ABSOLUTE

SPEC:RBW:ENTR:MODE Absolute

## SPECTrum:RBW:ENTRy:MODE?

Queries the bandwidth entry mode; absolute or relative



## Result Parameter

Parameter	Type	Description
Value	Enum	RBW entry (Absolute, Relative)

## Examples

//Queries the bandwidth entry

SPEC:RBW:ENTR:MODE?

ABSOLUTE

## SPECTrum:RBW:FILTter:TYPE

Sets the bandwidth filter type

## Command Parameter

Parameter	Type	Description
Value	Enum	Filter type (Normal, EMC)

## Examples

//Sets the bandwidth filter type to NORMAL

SPEC:RBW:FILT:TYPE Normal

## SPECTrum:RBW:FILTter:TYPE?

Queries the bandwidth filter type

## Result Parameter

Parameter	Type	Description
Value	Enum	Filter type (Normal, EMC)

## Examples

//Queries the bandwidth filter type

SPEC:RBW:FILT:TYPE?

NORMAL

## SPECTrum:RBW:LIST?

Queries a list of possible RBW settings

## Result Parameter

Parameter	Type	Description
List	FlexArray[Double]	List of RBWs

## Examples

//Queries a list of RBWs

SPEC:RBW:LIST?

800000,400000,200000,100000,50000,25000,12500,6250

### SPECTrum:SCAN:COUNT

Sets the spectrum scan count

#### Command Parameter

Parameter	Type	Description
Value	ULong	Scan count

#### Examples

//Sets the scan count to 50

SPEC:SCAN:COUN 50

### SPECTrum:SCAN:COUNT?

Queries the current scan count

#### Result Parameter

Parameter	Type	Description
Value	ULong	Scan count

#### Examples

//Queries the current scan count

SPEC:SCAN:COUN?

50

### SPECTrum:SCAN:NUMBer?

Queries the current scan number

#### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

#### Examples

//Queries the current scan number

SPEC:SCAN:NUMB?

1814

### SPECTrum:TRACe:DETEctor<Number>

Sets the detector configuration. Only the 3rd detector can be configured.

#### Suffixes

Parameter	Type	Description
Number	UShort	Detector Number = 3

## Command Parameter

Parameter	Type	Description
Mode	Enum	Detector1 (PPk), Detector2 (RMS), Detector3 (MPk, Smp, Avg)

## Examples

//Sets detector 3 to minus peak

SPEC:TRAC:DET3 MPk

## SPECTrum:TRACe:DETEctor<Number>?

Queries the detector configuration. Only the 3rd detector can be configured

## Suffixes

Parameter	Type	Description
Number	UShort	Detector Number = 3

## Result Parameter

Parameter	Type	Description
Mode	Enum	Detector1 (PPk), Detector2 (RMS), Detector3 (MPk, Smp, Avg)

## Examples

//Queries the detector mode for detector 3

SPEC:TRAC:DET3?

MPk

## SPECTrum:TRACe:ENABLE

Enables the displayed detectors and traces

## Command Parameter

Parameter	Type	Description
Name	Enum	All, RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS
State	Bool	Trace state

## Examples

//Enables AvR trace

SPEC:TRAC:ENAB AvR,ON

## SPECTrum:TRACe:ENABLE?

Queries the corresponding trace state

### Query Parameter

Parameter	Type	Description
Name	Enum	All, RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

### Result Parameter

Parameter	Type	Description
State	Bool	Trace state

### Examples

//Queries AvR trace state

SPEC:TRAC:ENAB? AvR

1

## SPECTrum:TRACe:INFinite

Sets the infinite min max trace state

### Command Parameter

Parameter	Type	Description
Value	Bool	Infinite state

### Examples

//Enables infinite min max trace

SPEC:TRAC:INF ON

## SPECTrum:TRACe:INFinite?

Queries the infinite min max trace state

### Result Parameter

Parameter	Type	Description
Value	Bool	Infinite state

### Examples

//Queries infinite state

SPEC:TRAC:INF?

1

## SPECTrum:TRACe:LIST?

Queries a list of selected display traces

### Result Parameter

Parameter	Type	Description
List	FlexArray[Enum]	Selected display traces

## Examples

//Queies all selected display traces

SPEC:TRAC:LIST?

RMS,PPk,AvR

## SPECTrum:TSTamp?

Queries the spectrum data timestamp

### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

## Examples

//Queries the spectrum data timestamp

SPEC:TST?

0,1533056334,809341586

### Narda Command Reference - MARKer

Marker subsystem

#### MARKer:FXD:FREQuency

Sets the frequency of fixed marker

##### Command Parameter

Parameter	Type	Description
Value	Frequency	Fixed Marker frequency (Hz, kHz, MHz, GHz)

##### Examples

//Sets fixed marker frequency to 52.2 MHz

MARK:FXD:FREQ 52.2e6

#### MARKer:FXD:FREQuency?

Queries the frequency of fixed marker

##### Result Parameter

Parameter	Type	Description
Value	Frequency	Marker frequency in Hz

##### Examples

//Queries the fixed marker frequency

MARK:FXD:FREQ?

52200000

#### MARKer:FXD:TIME

Sets the time position of fixed marker

##### Command Parameter

Parameter	Type	Description
Value	Timespan	Fixed Marker time (h, m, s, ms)

##### Examples

//Sets fixed marker time position to -15 s

MARK:FXD:TIME -15

//Sets fixed marker time position to -5 ms

MARKer:FXD:TIME -5 ms

#### MARKer:FXD:TIME?

Queries the time position of fixed marker

### Result Parameter

Parameter	Type	Description
Value	Timespan	Time position of fixed Marker in seconds

### Examples

//Queries the fixed marker time postion

MARK:FXD:TIME?

-15

### MARKer:FXD:VALue

Sets the value of fixed marker

### Command Parameter

Parameter	Type	Description
Value	Unit	Fixed marker value in actual unit (dBm,...)

### Examples

//Sets the fixed marker value to 10 in actual unit

MARK:FXD:VAL 10

//Sets the fixed marker value to 10 dBm

MARKer:FXD:VALue 10 dBm

### MARKer:FXD:VALue?

Queries the value of fixed marker

### Result Parameter

Parameter	Type	Description
Value	Unit	Fixed marker value in actual unit

### Examples

//Queries the fixed marker value

MARK:FXD:VAL?

10.00

### MARKer:SPECtrum:DATA:ALL?

Queries all marker values for spectrum. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).

### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

## Command Reference Guide

### Command Descriptions

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
MarkerID	Enum	Optional: M1, M2, D2, M3, D3, M4, D4, ... , M8, D8
MarkerElements	ULong	Number of MarkerID Elements
TraceName	Enum	RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS
Frequency	Double	Marker frequency in Hz
Time	Double	Marker time in seconds
Level	Float	Marker power level in actual unit
CPowerFunction	Enum	Optional: CHANNEL_POWER
CPower	Unit	Channel Power
NoiseFunction	Enum	Optional: NOISE
Noise	Unit	Noise
OcBwFunction	Enum	Optional: OCCUPIED_BANDWIDTH
OcBW	Frequency	Occupied Bandwidth value
OcBWFmid	Frequency	Middle Frequency
OcBWCPower	Unit	Channel Power
TrFunction	Enum	Optional: TRANSMITTER
TrName	String	Transmitter name
TrType	String	Transmitter type
TrComment	String	Transmitter comment

#### Examples

// Marker 1 with Marker Function NOISE, Marker 2 with Marker Function CHANNEL POWER, Marker 3 without function

MARK:SPEC:DATA:ALL?

0,1551351924,765450087,106,M1,6,RMS,100000000,0,-78.23,NOISE,-121.42,M2,6,RMS,104400000,0,-55.47,CHANNEL\_POWER,-50.91,M3,4,RMS,98000000,0,-47.77

#### MARKer<Index>:SPECtrum:DATA:FREQuency?

Queries the marker or deltamarker frequency data

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Result Parameter

Parameter	Type	Description
Data	Double	Marker frequency in Hz



## Examples

//Marker 3 frequency

MARK3:SPEC:DATA:FREQ?

60000000

## MARKer<Index>:SPECtrum:DATA:LEVel?

Queries marker or deltamarker level data

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Data	Float	Marker power level in actual unit

## Examples

//Marker 3 power level

MARK3:SPEC:DATA:LEV?

-0.85

## MARKer<Index>:SPECtrum:DATA:PEAK:STATe?

Queries the marker peak state. PEAK:STATe only works in HOLD or STOP mode or with track peak enabled (PEAK:TRACK ON.)

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Value	Bool	1 = peak was found, 0 = no peak was found

## Examples

//Marker 3 peak state

MARK3:SPEC:DATA:PEAK:STAT?

0

## MARKer<Index>:SPECtrum:DATA:TIME?

Queries the marker or deltamarker time data

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Data	Double	Marker time in seconds

### Examples

//Time position of marker 2

MARK2:SPEC:DATA:TIME?

0

## MARKer<Index>:SPECtrum:ENABLE

Marker enable or disable

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Command Parameter

Parameter	Type	Description
Value	Bool	Enable state

### Examples

//Disables marker 3

MARK3:SPEC:ENAB OFF

## MARKer<Index>:SPECtrum:ENABLE?

Queries the current state of a marker

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Value	Bool	Enable state

### Examples

//Queries the marker 1 enable state

MARK1:SPEC:ENAB?

1

## MARKer<Index>:SPECtrum:FREQuency

Marker frequency (see also MARKer<Index>:SPECtrum:DATA:FREQuency?)

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Command Parameter

Parameter	Type	Description
Value	Frequency	Marker frequency (Hz, kHz, MHz, GHz)

### Examples

//Sets marker 2 frequency to 52.2 MHz

MARK2:SPEC:FREQ 52.2e6

## MARKer<Index>:SPECtrum:FREQuency?

Queries the current frequency for a marker

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Value	Frequency	Marker frequency in Hz

### Examples

//Queries the current frequency for marker 1

MARK1:SPEC:FREQ?

52200000

## MARKer<Index>:SPECtrum:FREQuency:LINK

Marker frequency link

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Command Parameter

Parameter	Type	Description
LinkIndex	ULong	Frequency link Marker Index (1...8). If Marker Index equals Link Index, Frequency link is disabled.

### Examples

//Disables frequency link for marker 4

MARK4:SPEC:FREQ:LINK 4

//Sets marker 8 as frequency link for marker 3

MARK3:SPEC:FREQ:LINK 8

## MARKer<Index>:SPECtrum:FREQuency:LINK?

Queries the marker frequency link to another marker

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
LinkIndex	ULong	Frequency link Marker Index (1...8)

### Examples

//Queries the marker which is linked (in frequency) with marker 2

MARK2:SPEC:FREQ:LINK?

1

## MARKer<Index>:SPECtrum:FREQuency:LINK:OFFSet

Marker frequency link offset

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

### Command Parameter

Parameter	Type	Description
Value	Frequency	Marker frequency (Hz, kHz, MHz, GHz)

## Examples

//Sets marker 3 frequency link offset to 10 MHz

MARK3:SPEC:FREQ:LINK 10e6

## MARKer<Index>:SPECtrum:FREQuency:LINK:OFFSet?

Queries the frequency link offset

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

### Result Parameter

Parameter	Type	Description
Value	Frequency	Marker frequency in Hz

## Examples

//Queries the frequency link offset for marker 2

MARKer2:SPECtrum:FREQuency:LINK:OFFSet?

10000000

## MARKer<Index>:SPECtrum:FUNCtion

Sets the marker function

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Command Parameter

Parameter	Type	Description
Function	String	Function name e.g. 'CHANNEL_POWER', 'NOISE', 'OCCUPIED_BANDWIDTH', 'TRANSMITTER' or empty string " to disable function

## Examples

//Disables function for marker 3

MARK3:SPEC:FUNCtion "

//Sets channel power function for marker 1

MARKer1:SPECtrum:FUNCtion 'CHANNEL\_POWER'

//Sets noise function for marker 2

MARK2:SPEC:FUNCtion 'NOISE'

### MARKer<Index>:SPECtrum:FUNCtion?

Queries the channel power function

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Result Parameter

Parameter	Type	Description
Function	String	Function name

#### Examples

//Queries the channel power function for marker 1

MARK1:SPEC:FUNC?

"CHANNEL\_POWER"

### MARKer:SPECtrum:FUNCtion:CPOWer:CBW

CBW for marker function: Channel Power

#### Command Parameter

Parameter	Type	Description
Value	Frequency	CBW frequency (Hz, kHz, MHz, GHz)

#### Examples

//Sets CBW 100 kHz for channel power function of all marker

MARK:SPEC:FUNC:CPOW:CBW 100e3

### MARKer:SPECtrum:FUNCtion:CPOWer:CBW?

Queries the CBW

#### Result Parameter

Parameter	Type	Description
Value	Frequency	CBW frequency in Hz

#### Examples

//Queries the CBW for channel power function of all marker

MARK:SPEC:FUNC:CPOW:CBW?

100000

### MARKer<Index>:SPECtrum:FUNCtion:CPOWer:DATA?

Queries data for marker function: Channel Power

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Result Parameter

Parameter	Type	Description
Data	Unit	Marker function data in actual unit

## Examples

//Marker 3 function data

MARK3:SPEC:FUNC:CPOW:DATA?

-86.61

## MARKer<Index>:SPECtrum:FUNCtion:NOISe:DATA?

Queries data for marker or deltamarker function: Noise

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Result Parameter

Parameter	Type	Description
Data	Unit	Marker function data in actual unit

## Examples

//Marker 3 function data

MARK3:SPEC:FUNC:NOIS:DATA?

-146.54

## MARKer:SPECtrum:FUNCtion:NOISe:NBW

NBW for marker function: Noise

## Command Parameter

Parameter	Type	Description
Value	Frequency	NBW frequency (Hz, kHz, MHz, GHz)

## Examples

//Sets NBW 100 kHz for noise function of all marker

MARK:SPEC:FUNC:NOIS:NBW 100e3

### MARKer:SPECTrum:FUNCTION:NOISe:NBW?

Queries the NBW

#### Result Parameter

Parameter	Type	Description
Value	Frequency	NBW frequency in Hz

#### Examples

//Queries the NBW for noise function of all marker

MARK:SPEC:FUNC:NOIS:NBW?

1000000

### MARKer:SPECTrum:FUNCTION:NOISe:NBW:AUTO

NBW auto coupling for marker function: Noise

#### Command Parameter

Parameter	Type	Description
Value	Bool	Coupling state

#### Examples

//Sets NBW auto coupling to OFF for noise function of all marker

MARK:SPEC:FUNC:NOIS:NBW:AUTO OFF

### MARKer:SPECTrum:FUNCTION:NOISe:NBW:AUTO?

Queries the NBW auto coupling state

#### Result Parameter

Parameter	Type	Description
Value	Bool	Coupling state

#### Examples

//Queries the NBW auto coupling state for noise function of all marker

MARK:SPEC:FUNC:NOIS:NBW:AUTO?

0

### MARKer<Index>:SPECTrum:FUNCTION:OCBW:DATA?

Queries data for marker function: OCCUPIED\_BANDWIDTH

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)



## Result Parameter

Parameter	Type	Description
OcBW	Frequency	Occupied Bandwidth value
OcBWFmid	Frequency	Middle Frequency
OcBWCPower	Unit	Channel Power

## Examples

//Marker 3 function data

MARK3:SPEC:FUNC:OCBW:DATA?

155663,446019301,-21.37

//Marker 2 function data

MARKer2:SPECtrum:FUNCTION:OCBW:DATA?

155723,446019276,-21.10

## MARKer:SPECtrum:FUNCTION:OCBW:MODE

Mode for marker function: OCCUPIED\_BANDWIDTH

## Command Parameter

Parameter	Type	Description
Value	Enum	Occupied bandwith mode (% , dB)

## Examples

//Sets occupied bandwith mode to dB

MARK:SPEC:FUNC:OCBW:MODE dB

## MARKer:SPECtrum:FUNCTION:OCBW:MODE?

Mode for marker function: OCCUPIED\_BANDWIDTH

## Result Parameter

Parameter	Type	Description
Value	Enum	Occupied bandwith mode

## Examples

//Queries occupied bandwith mode

MARK:SPECtrum:FUNC:OCBW:MODE?

XdB

## MARKer:SPECtrum:FUNCTION:OCBW:NTRials

Sets the NTRials (number of trials) for marker function: OCCUPIED\_BANDWIDTH

### Command Parameter

Parameter	Type	Description
Value	UShort	Occupied bandwidth NTrials parameter (1...1000)

### Examples

//Sets occupied bandwidth NTrials value to 80

MARK:SPEC:FUNC:OCBW:NTR 80

### MARKer:SPECTrum:FUNctio:OCBW:NTRials?

NTrials (number of trials) for marker function: OCCUPIED\_BANDWIDTH

### Result Parameter

Parameter	Type	Description
Value	UShort	Occupied bandwidth NTrials parameter

### Examples

//Queries occupied bandwidth NTrials value

MARK:SPEC:FUNC:OCBW:NTR?

80

### MARKer:SPECTrum:FUNctio:OCBW:PERCent

Sets the % for marker function: OCCUPIED\_BANDWIDTH

### Command Parameter

Parameter	Type	Description
Value	Float	Occupied bandwidth % parameter (50 ... 99.99)

### Examples

//Sets occupied bandwidth percent value to 80

MARKer:SPECTrum:FUNctio:OCBW:PERCent 80.0

### MARKer:SPECTrum:FUNctio:OCBW:PERCent?

Queries the % for marker function: OCCUPIED\_BANDWIDTH

### Result Parameter

Parameter	Type	Description
Value	Float	Occupied bandwidth % parameter

## Examples

//Queries occupied bandwidth percent value

MARK:SPECTrum:FUNC:OCBW:PERC?

80

## MARKer:SPECTrum:FUNCtion:OCBW:THReshold

Sets the threshold for marker function: OCCUPIED\_BANDWIDTH

### Command Parameter

Parameter	Type	Description
Value	Unit	Threshold level in actual unit or units of: No antenna connected: dBm, dBV, dBmV, dBuV  With antenna connected: dBm, dBV, dBmV, dBuV, W_m2, W_cm2, A_m, V_m, dBA_m, dBV_m, dBmV_m, dBuV_m

## Examples

//Sets occupied bandwidth threshold level to 0 dBm

MARK:SPEC:FUNC:OCBW:THR 0 dBm

## MARKer:SPECTrum:FUNCtion:OCBW:THReshold?

Queries the threshold for marker function: OCCUPIED\_BANDWIDTH

### Result Parameter

Parameter	Type	Description
Value	Unit	Threshold level in actual unit

## Examples

//Queries occupied bandwidth threshold level

MARK:SPEC:FUNC:OCBW:THR?

0.00

## MARKer:SPECTrum:FUNCtion:OCBW:XDB

XdB for marker function: OCCUPIED\_BANDWIDTH

### Command Parameter

Parameter	Type	Description
Value	Float	Occupied bandwidth XdB parameter (-60...1)

### Examples

//Sets occupied bandwidth XdB value to -10 dB

MARK:SPEC:FUNC:OCBW:XDB -10.0

### MARKer:SPECTrum:FUNCtion:OCBW:XDB?

XdB for marker function: OCCUPIED\_BANDWIDTH

#### Result Parameter

Parameter	Type	Description
Value	Float	Occupied bandwidth XdB parameter

### Examples

//Queries occupied bandwidth XdB

MARK:SPEC:FUNC:OCBW:XDB?

-10

### MARKer<Index>:SPECTrum:FUNCtion:TRANsmitter:DATA?

Queries the data for Marker function: TRANSMITTER

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Result Parameter

Parameter	Type	Description
Name	String	Transmitter name
Type	String	Transmitter type
Comment	String	Transmitter comment

### Examples

//Marker 2 function data

MARKer2:SPECTrum:FUNCtion:TRANsmitter:DATA?

"BIG FM","FM Radio","Deutschlands biggste Beats"

//Marker 3 function data

MARK3:SPEC:FUNC:TRAN:DATA?

"SWR 4 BW","FM Radio",""

### MARKer:SPECTrum:LIST?

List of enabled markers

## Result Parameter

Parameter	Type	Description
List	FlexArray[Enum]	Enabled marker list

## Examples

//Queries the enabled markers

MARK:SPEC:LIST?

M1,M2

## MARKer<Index>:SPECtrum:PEAK

Marker peak search (time or frequency domain)

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Examples

//Marker 2 peak search

MARK2:SPEC:PEAK

## MARKer<Index>:SPECtrum:PEAK:LEFT

Marker left peak search (frequency domain)

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Examples

//Marker 2 left peak search

MARK2:SPEC:PEAK:LEFT

## MARKer<Index>:SPECtrum:PEAK:LOWer

Marker lower peak search (time domain)

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Examples

//Marker 2 lower peak search

MARK2:SPEC:PEAK:LOW

### MARKer<Index>:SPECtrum:PEAK:NEXT

Marker next peak search (time or frequency domain)

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Examples

//Marker 2 next peak search

MARK2:SPEC:PEAK:NEXT

### MARKer<Index>:SPECtrum:PEAK:RIGHT

Marker right peak search (frequency domain)

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Examples

//Marker 2 right peak search

MARK2:SPEC:PEAK:RIGHT

### MARKer<Index>:SPECtrum:PEAK:UPPer

Marker upper peak search (time domain)

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Examples

//Marker 2 upper peak search

MARK2:SPEC:PEAK:UPP

### MARKer<Index>:SPECtrum:REFerence

Reference marker for a delta marker

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

## Command Parameter

Parameter	Type	Description
RefIndex	ULongEnum	Reference Marker Index (0...8, FXD). If RefIndex equals 0, Reference is disabled.

## Examples

//Disables reference for marker 4

MARK4:SPEC:REF 0

//Sets marker 1 as reference for delta marker 2

MARKer2:SPECtrum:REFerence 1

//Sets FXD marker as reference for delta marker 3

MARK3:SPEC:REF FXD

## MARKer<Index>:SPECtrum:REFerence?

Queries the corresponding reference marker

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

## Result Parameter

Parameter	Type	Description
RefIndex	ULongEnum	Reference Marker Index (0...8, FDX)

## Examples

//Queries the reference marker for delta marker 2

MARK2:SPEC:REF?

1

## MARKer:SPECtrum:SEARch:AUTO:PEAK:ENABLE

'Auto Peak Search for M1' for marker

## Command Parameter

Parameter	Type	Description
Value	Bool	'Auto Peak Search for M1'

## Examples

//Sets the searchlimit enable state to OFF

MARK:SPEC:SEAR:AUTO:PEAK:ENAB 0

## MARKer:SPECtrum:SEARch:AUTO:PEAK:ENABLE?

Queries the searchlimit enable state

### Result Parameter

Parameter	Type	Description
Value	Bool	'Auto Peak Search for M1'

### Examples

//Queries the searchlimit enable state

MARK:SPEC:SEAR:AUTO:PEAK:ENAB?

1

## MARKer:SPECTrum:SEARch:FREQuency:LOWer

Lower frequency searchlimit for marker and peaktable

### Command Parameter

Parameter	Type	Description
Value	Frequency	Frequency searchlimit (Hz, kHz, MHz, GHz)

### Examples

//Sets the lower frequency searchlimit to 100 MHz

MARK:SPEC:SEAR:FREQ:LOW 100 MHz

## MARKer:SPECTrum:SEARch:FREQuency:LOWer?

Queries the lower frequency searchlimit

### Result Parameter

Parameter	Type	Description
Value	Frequency	Frequency searchlimit in Hz

### Examples

//Queries the lower frequency searchlimit

MARK:SPEC:SEAR:FREQ:LOW?

100000000

## MARKer:SPECTrum:SEARch:FREQuency:UPPer

Upper frequency searchlimit for marker and peaktable

### Command Parameter

Parameter	Type	Description
Value	Frequency	Frequency searchlimit (Hz, kHz, MHz, GHz)

### Examples

//Sets the upper frequency searchlimit to 400 MHz

MARK:SPEC:SEAR:FREQ:UPP 400 MHz



## MARKer:SPECTrum:SEARch:FREQuency:UPPer?

Queries the upper frequency searchlimit

### Result Parameter

Parameter	Type	Description
Value	Frequency	Frequency searchlimit in Hz

### Examples

//Queries the upper frequency searchlimit

MARK:SPEC:SEAR:FREQ:UPP?

400000000

## MARKer:SPECTrum:SEARch:LIMits:ENABLE

Sets the 'Use Search Limits' enable state for marker and peaktable

### Command Parameter

Parameter	Type	Description
Value	Bool	'Use Search Limits'

### Examples

//Sets the searchlimit enable state to OFF

MARK:SPEC:SEAR:LIM:ENAB 0

## MARKer:SPECTrum:SEARch:LIMits:ENABLE?

Queries the searchlimit enable state

### Result Parameter

Parameter	Type	Description
Value	Bool	'Use Search Limits'

### Examples

//Queries the searchlimit enable state

MARK:SPEC:SEAR:LIM:ENAB?

1

## MARKer:SPECTrum:SEARch:LOEXclude:ENABLE

'Exclude LO' searchlimit for marker and peaktable

### Command Parameter

Parameter	Type	Description
Value	Bool	'Exclude LO'

### Examples

//Sets the searchlimit enable state to OFF

MARK:SPEC:SEAR:LOEX:ENAB 0

## MARKer:SPECTrum:SEARch:LOEXclude:ENABLE?

Queries the searchlimit enable state

### Result Parameter

Parameter	Type	Description
Value	Bool	'Exclude LO'

### Examples

//Queries the searchlimit enable state

MARK:SPEC:SEAR:LOEX:ENAB?

0

## MARKer:SPECTrum:SEARch:PEAK:EXCursion

Sets the peak excursion for marker and peaktable

### Command Parameter

Parameter	Type	Description
Value	Unit	Peak Excursion (dB)

### Examples

//Sets the peak excursion to 10 dB

MARK:SPEC:SEAR:PEAK:EXC 10 dB

## MARKer:SPECTrum:SEARch:PEAK:EXCursion?

Queries the peak excursion

### Result Parameter

Parameter	Type	Description
Value	Unit	Peak Excursion in dB

### Examples

//Queries the peak excursion

MARK:SPEC:SEAR:PEAK:EXC?

10

## MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABLE

'Peak Excursion' searchlimit for marker and peaktable

## Command Parameter

Parameter	Type	Description
Value	Bool	'Peak Excursion'

## Examples

//Sets the peak excursion enable state to ON

MARK:SPEC:SEAR:PEAK:EXC:ENAB 0

## MARKer:SPECTrum:SEARch:PEAK:EXCursion:ENABLE?

Queries the peak excursion enable state

## Result Parameter

Parameter	Type	Description
Value	Bool	'Peak Excursion'

## Examples

//Queries the peak excursion enable state

MARK:SPEC:SEAR:PEAK:EXC:ENAB?

1

## MARKer:SPECTrum:SEARch:SGRam:RANGe

Sets the spectrogram searchrange

## Command Parameter

Parameter	Type	Description
Value	Enum	Spectrogram searchrange (All, Visible)

## Examples

//Sets spectrogram searchrange to VISIBLE

MARK:SPEC:SEAR:SGR:RANG VISIBLE

## MARKer:SPECTrum:SEARch:SGRam:RANGe?

Queries the spectrogram searchrange

## Result Parameter

Parameter	Type	Description
Value	Enum	Spectrogram searchrange (All, Visible)

## Examples

//Queries the spectrogram searchrange

MARK:SPEC:SEAR:SGR:RANG?

VISIBLE

### MARKer:SPECTrum:SEARch:THReshold

Threshold searchlimit for marker and peaktable

#### Command Parameter

Parameter	Type	Description
Value	Unit	Threshold level in actual unit or units of: No antenna connected: dBm, dBV, dBmV, dBuV  With antenna connected: dBm, dBV, dBmV, dBuV, W_m2, W_cm2, A_m, V_m, dBA_m, dBV_m, dBmV_m, dBuV_m

#### Examples

//Sets the threshold searchlimit to -80 dBm

MARK:SPEC:SEAR:THR -80 dBm

//Sets the threshold searchlimit to -100 in actual unit

MARK:SPEC:SEAR:THR -100

### MARKer:SPECTrum:SEARch:THReshold?

Queries the threshold searchlimit

#### Result Parameter

Parameter	Type	Description
Value	Unit	Threshold level in actual unit

#### Examples

//Queries the threshold searchlimit

MARK:SPEC:SEAR:THR?

-80.00

### MARKer:SPECTrum:SEARch:TRACk:PEAKs:ENABLE

'Track Peaks' for marker

#### Command Parameter

Parameter	Type	Description
Value	Bool	'Track Peaks'

#### Examples

//Sets the searchlimit enable state to OFF

MARK:SPEC:SEAR:TRAC:PEAK:ENAB 0

## MARKer:SPECTrum:SEARch:TRACk:PEAKs:ENABLE?

Queries the searchlimit enable state

### Result Parameter

Parameter	Type	Description
Value	Bool	'Track Peaks'

### Examples

//Queries the searchlimit enable state

MARK:SPEC:SEAR:TRAC:PEAK:ENAB?

0

## MARKer<Index>:SPECTrum:TIME

Time position of a marker (see also MARKer<Index>:SPECTrum:DATA:TIME?)

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Command Parameter

Parameter	Type	Description
Value	Timespan	Marker time (h, m, s, ms)

### Examples

//Sets marker 2 time position to -7 ms

MARK2:SPEC:TIME -7e-3

## MARKer<Index>:SPECTrum:TIME?

Queries marker time position

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Value	Timespan	Time position of the Marker in seconds

### Examples

//Queries marker 1 time position

MARK1:SPEC:TIME?

0

### MARKer<Index>:SPECtrum:TIME:LINK

Marker time link

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Command Parameter

Parameter	Type	Description
LinkIndex	ULong	Time link Marker Index (1...8). If Marker Index equals Link Index, Time link is disabled.

#### Examples

//Disables time link for marker 4

MARK4:SPEC:TIME:LINK 4

//Sets marker 8 as time link for marker 3

MARK3:SPEC:TIME:LINK 8

### MARKer<Index>:SPECtrum:TIME:LINK?

Queries the time link marker

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

#### Result Parameter

Parameter	Type	Description
LinkIndex	ULong	Time link Marker Index (1...8)

#### Examples

//Queries the time link marker for marker 2

MARK2:SPEC:TIME:LINK?

2

### MARKer<Index>:SPECtrum:TIME:LINK:OFFSet

Marker time link offset. A time offset can only be set if the selected task is an RT Spectrum task.

#### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

## Command Parameter

Parameter	Type	Description
Value	Timespan	Offset time (h, m, s, ms)

## Examples

//Sets time link offset of marker 2 to 5 ms

MARK2:SPEC:TIME:LINK:OFFS 5 ms

## MARKer<Index>:SPECtrum:TIME:LINK:OFFSet?

Queries the time link offset

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

## Result Parameter

Parameter	Type	Description
Value	Timespan	Offset time in seconds

## Examples

//Queries the time link offset of marker 2

MARK2:SPEC:TIME:LINK:OFFS?

0

## MARKer<Index>:SPECtrum:TRACe

Selects the active trace for a marker. Only traces enabled by SPECtrum:TRACe:ENABLE are available.

## Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

## Command Parameter

Parameter	Type	Description
Name	Enum	RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

## Examples

//Sets RMS as active trace for marker 2

MARK2:SPEC:TRAC RMS

## MARKer<Index>:SPECtrum:TRACe?

Queries the active trace of a marker

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (1...8)

### Result Parameter

Parameter	Type	Description
Name	Enum	RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

### Examples

//Queries the current active trace for marker 1

MARK1:SPEC:TRAC?

AvR

## MARKer<Index>:SPECtrum:TYPE

Sets the Marker type

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

### Command Parameter

Parameter	Type	Description
Type	Enum	Type (NORMAL,DELTA)

### Examples

//Configures Marker 3 as delta Marker

MARK3:SPEC:TYPE DELTA

## MARKer<Index>:SPECtrum:TYPE?

Queries the configuration of a marker, normal or delta

### Suffixes

Parameter	Type	Description
Index	UShort	Marker Index (2...8)

### Result Parameter

Parameter	Type	Description
Type	Enum	Type (NORMAL,DELTA)



### Examples

//Queries the configuration of marker 2

MARK2:SPEC:TYPE?

DELTA

## Narda Command Reference - LEVelmeter

Levelmeter subsystem

### LEVelmeter:CBW

Sets the channel bandwidth for levelmeter

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth (Hz, kHz, MHz, GHz)

#### Examples

//Sets the bandwith in kHz

LEV:CBW 100 kHz

### LEVelmeter:CBW?

Queries the channel bandwidth for levelmeter

#### Result Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth in Hz

#### Examples

//Queries the bandwith

LEV:CBW?

100000

### LEVelmeter:CBW:FILTer:TYPE

Sets the levelmeter CBW filter type

#### Command Parameter

Parameter	Type	Description
Value	Enum	Filter type (Channel, MIL, CISPR)

#### Examples

//Sets the cbw filter type to Channel

LEVelmeter:CBW:FILTer:TYPE Channel

### LEVelmeter:CBW:FILTer:TYPE?

Queries the levelmeter CBW filter type

**Result Parameter**

Parameter	Type	Description
Value	Enum	Filter type (Channel, MIL, CISPR)

**Examples**

```
//Queries the cbw filter type
```

```
LEV:CBW:FILT:TYPE?
```

```
CHANNEL
```

**LEVelmeter:CBW:LIST?**

Queries a list of possible CBW settings

**Result Parameter**

Parameter	Type	Description
List	FlexArray[Double]	List of CBWs

**Examples**

```
//Queries a list of CBWs
```

```
LEV:CBW:LIST?
```

```
40000000,38400000,32000000,30000000,25600000,25000000,24000000,20000000,19200000,16000000,15000000,12800000,12500000,12000000,10000000,9600000,8000000,7500000,6400000,6250000,6000000,5120000,5000000,4800000,4000000,3840000,3200000,3000000,2560000,2500000,2400000,2000000,1920000,1600000,1500000,1280000,1250000,1200000,1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000,300000,256000,250000,240000,200000,192000,160000,150000,128000,125000,120000,100000,96000,80000,75000,64000,62500,60000,51200,50000,48000,40000,38400,32000,30000,25600,25000,24000,20000,19200,16000,15000,12800,12500,12000,10000,9600,9000,8000,7500,6400,6250,6000,5120,5000,4800,4000,3840,3200,3100,3000,2700,2560,2500,2400,2100,2000,1920,1600,1500,1280,1250,1200,1000,960,800,750,640,625,600,512,500,480,400,384,320,300,256,250,240,200,192,160,150,128,125,120,100,50,25
```

**LEVelmeter:CBW:OVERsampling**

Sets the levelmeter oversampling state

**Command Parameter**

Parameter	Type	Description
Value	Bool	Oversampling

**Examples**

```
//Disables oversampling
```

```
LEV:CBW:OVER OFF
```

**LEVelmeter:CBW:OVERsampling?**

Queries the levelmeter oversampling state

### Result Parameter

Parameter	Type	Description
Value	Bool	Oversampling

### Examples

//Queries the levelmeter oversampling state

LEV:CBW:OVER?

1

## LEVelmeter:DATA:AFC?

Queries AFC data values

### Result Parameter

Parameter	Type	Description
FCorr	Frequency	FTune correction in Hz
FOffset	Frequency	Offset frequency in Hz

### Examples

//Queries AFC data

LEV:DATA:AFC?

0,-1959

## LEVelmeter:DATA:ALL?

Queries all levelmeter detector and handle values. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).

### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

## Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
CompassID	Enum	Optional: COMPASS
CompassElements	ULong	Number of CompassID Elements
Azimuth	Float	Azimuth in degree.
Elevation	Float	Elevation in degree.
Roll	Float	Roll in degree.
AfcID	Enum	Optional: AFC
AfcElements	ULong	Number of AfcID Elements
AfcFCorr	Frequency	FTune correction in Hz
AfcFOffset	Frequency	Offset frequency in Hz
DetectorID	Enum	Optional: PPk, CPk, RMS, CRMS, MPk, Smp, Avg, CAvg
DetectorElements	ULong	Number of DetectorID Elements
Overdriven	Bool	Overdriven flag
NotRealtime	Bool	NotRealtime flag
DetectorValue	Float	Detector value
TraceValue	Float	Detector Trace min/max value
ModDetectorID	Enum	Optional: AM_PPk, AM_MPk, AM_PPDIV2, AM_RMS, FM_PPk, FM_MPk, FM_PPDIV2, FM_RMS, PM_PPk, PM_MPk, PM_PPDIV2, PM_RMS
ModDetectorElements	ULong	Number of ModDetectorID Elements
ModOverdriven	Bool	Overdriven flag
ModNotRealtime	Bool	NotRealtime flag
ModDetectorValue	Float	Detector value
ModTraceValue	Float	Detector Trace min/max value

## Examples

//Queries all activated detectors of levelmeter (no handle connected, 3 level detectors and 4 FM modulation detectors )

LEV:DATA:ALL?

0,1551352285,20747734,2160,AFC,2,0,395349,PPk,4,0,0,-67.44,-67.44,RMS,4,0,0,-73.29,-73.29,Avg,4,0,0,-73.64,-73.64,FM\_PPk,4,0,0,244645,244645,FM\_MPk,4,0,0,1035296,1035296,FM\_PPdiv2,4,0,0,639971,639971,FM\_RMS,4,0,0,87840,87840

//Queries all activated detectors of levelmeter (with handle connected, 1 detector)

LEV:DATA:ALL?

0,1533059558,957863188,608,COMPASS,3,150.7,1.2,1.3,PPk,4,0,0,-70.71,-70.71

### LEVelmeter:DATA:DETEctor<Number>?

Queries the levelmeter detector value (PPk, CPk, RMS, CRMS, MPk, Smp, Avg, CAvg)

#### Suffixes

Parameter	Type	Description
Number	UShort	Detector Number 1...3

#### Result Parameter

Parameter	Type	Description
Value	Float	Detector value
ValueMinMax	Float	Detector min/max value

#### Examples

//Queries the levelmeter detector2 values

LEV:DATA:DET2?

-84.70,-79.03

### LEVelmeter:DATA:DETEctor:MODulation?

Queries the levelmeter modulation detector values (AM\_PPk, AM\_MPk, AM\_PPDIV2, AM\_RMS, FM\_PPk, FM\_MPk, FM\_PPDIV2, FM\_RMS, PM\_PPk, PM\_MPk, PM\_PPDIV2, PM\_RMS)

#### Result Parameter

Parameter	Type	Description
ModDetectorID	Enum	AM_PPk, AM_MPk, AM_PPDIV2, AM_RMS, FM_PPk, FM_MPk, FM_PPDIV2, FM_RMS, PM_PPk, PM_MPk, PM_PPDIV2, PM_RMS
ModDetectorElements	ULong	Number of Elements
ModDetectorValue	Float	Detector value
ModTraceValue	Float	Detector Trace min/max value

#### Examples

//Queries the levelmeter modulation detector values

LEV:DATA:DET:MOD?

AM\_RMS,2,0.00,0.00,FM\_RMS,2,0,0,PM\_RMS,2,0.00,0.00

### LEVelmeter:DATA:UPDate?

Update is available in RUN and HOLD mode (see DATA:UPDate Commands)

## Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong, Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

## Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.

## Examples

//Levelmeter data update with old scancount = 10 and timeout = 100ms

LEV:DATA:UPD? 10,100ms

1520

//Force Levelmeter data update because we do not have old scannumber

LEV:DATA:UPDate?

1305

//Levelmeter data update with old scannumber = 0

LEV:DATA:UPDate? 0

1389

//Force Levelmeter data update with timeout = 100ms

LEV:DATA:UPD? NaN,100ms

1453

## LEV:DETECTOR<Number>

Detector configuration

## Suffixes

Parameter	Type	Description
Number	UShort	Detector Number 1...3

## Command Parameter

Parameter	Type	Description
Mode	Enum	Detector1 (Off, PPk, CPk), Detector2 (Off, RMS, CRMS), Detector3 (Off, MPk, Smp, Avg, CAvg)

## Examples

//Sets detector 3 to minus peak

LEV:DET3 MPk

### LEVelmeter:DETEctor<Number>?

Queries the detector configuration

#### Suffixes

Parameter	Type	Description
Number	UShort	Detector Number 1...3

#### Result Parameter

Parameter	Type	Description
Mode	Enum	Detector1 (Off, PPk, CPk), Detector2 (Off, RMS, CRMS), Detector3 (Off, MPk, Smp, Avg, CAvg)

#### Examples

//Queries current mode for detector 3

LEV:DET3?

MPk

### LEVelmeter:DETEctor:INFinite

Sets the infinite min max detector state

#### Command Parameter

Parameter	Type	Description
Value	Bool	Infinite state

#### Examples

//Enables infinite min max detector

LEV:DET:INF ON

### LEVelmeter:DETEctor:INFinite?

Queries the infinite min max detector state

#### Result Parameter

Parameter	Type	Description
Value	Bool	Infinite state

#### Examples

//Queries infinite state

LEVelmeter:DETEctor:INFinite?

1

### LEVelmeter:DETEctor:MODulation

Modulation detector configuration



## Command Parameter

Parameter	Type	Description
Mode	Enum	Detector mode (Off, AM, FM, PM, PPK, MPk, PPdiv2, RMS)

## Examples

//Sets modulation detector to AM

LEV:DET:MOD AM

## LEVelmeter:DETEctor:MODulation?

Queries the modulation detector configuration

## Result Parameter

Parameter	Type	Description
Mode	Enum	Detector mode (Off, AM, FM, PM, PPK, MPk, PPdiv2, RMS)

## Examples

//Queries modulation detector configuration

LEV:DET:MOD?

AM

## LEVelmeter:FREQuency:TUNE

Sets the levelmeter tune frequency

## Command Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency (Hz, kHz, MHz, GHz)

## Examples

//Sets the tune frequency in MHz

LEV:FREQ:TUNE 1.2 MHz

## LEVelmeter:FREQuency:TUNE?

Queries the levelmeter tune frequency

## Result Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency in Hz

### Examples

//Queries the tune frequency

LEV:FREQ:TUNE?

1200000

## LEVelmeter:FREQuency:TUNE:CENTer:COUPling

Automatic coupling of Ftune and Fcent

### Command Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Sets the Ftune Fcent coupling to OFF

LEV:FREQ:TUNE:CENT:COUP OFF

## LEVelmeter:FREQuency:TUNE:CENTer:COUPling?

Queries the coupling state of Ftune and Fcent

### Result Parameter

Parameter	Type	Description
Value	Bool	Coupling state

### Examples

//Queries the Ftune Fcent coupling state

LEV:FREQ:TUNE:CENT:COUP?

0

## LEVelmeter:FREQuency:TUNE:STEP

Tune frequency step for levelmeter

### Command Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency step (Hz, kHz, MHz, GHz)

### Examples

//Sets the tune frequency step in MHz

LEV:FREQ:TUNE:STEP 1 MHz

## LEVelmeter:FREQuency:TUNE:STEP?

Queries the tune frequency step for levelmeter

## Result Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency step in Hz

## Examples

//Queries the tune frequency step

LEV:FREQ:TUNE:STEP?

1000000

## LEVelmeter:MEASurement:TIME

Sets the measurement time for levelmeter

## Command Parameter

Parameter	Type	Description
Value	Timespan	Measurement time (h, m, s, ms)

## Examples

//Sets the measurement time to 10ms

LEV:MEAS:TIME 10 ms

## LEVelmeter:MEASurement:TIME?

Queries the measurement time for levelmeter

## Result Parameter

Parameter	Type	Description
Value	Timespan	Measurement time in s

## Examples

//Queries the measurement time

LEV:MEAS:TIME?

0.01

## LEVelmeter:POST:AVG

Sets the post averaging time for levelmeter

## Command Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time (h, m, s, ms)

## Examples

//Sets the post averaging time to 10ms

LEV:POST:AVG 10 ms

### LEVelmeter:POST:AVG?

Queries the post averaging time

#### Result Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time in s

#### Examples

//Queries the post averaging time

LEV:POST:AVG?

0.01

### LEVelmeter:PRE:AVG

Sets the pre averaging time for levelmeter

#### Command Parameter

Parameter	Type	Description
Value	Timespan	Pre averaging time (h, m, s, ms), 0 = OFF

#### Examples

//Sets the pre averaging time to 10ms

LEV:PRE:AVG 10 ms

### LEVelmeter:PRE:AVG?

Queries the pre averaging time

#### Result Parameter

Parameter	Type	Description
Value	Timespan	Pre averaging time in s, 0 = OFF

#### Examples

//Queries the pre averaging time

LEV:PRE:AVG?

0.01

### LEVelmeter:SCAN:COUNT

Sets the levelmeter scan count

#### Command Parameter

Parameter	Type	Description
Value	ULong	Scan count

## Examples

//Sets the scan count to 50

LEV:SCAN:COUN 50

## LEVelmeter:SCAN:COUNT?

Queries the levelmeter scan count

### Result Parameter

Parameter	Type	Description
Value	ULong	Scan count

## Examples

//Queries the scan count

LEV:SCAN:COUN?

50

## LEVelmeter:SCAN:NUMBer?

Queries the levelmeter scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

## Examples

//Queries the current scan number

LEV:SCAN:NUMB?

121

## LEVelmeter:TSTamp?

Queries levelmeter data timestamp

### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

## Examples

//Queries the levelmeter data timestamp

LEV:TST?

0,1533060782,779856660

### Narda Command Reference - PEAKtable

Peaktable subsystem, only includes parameters not included in Spectrum or Marker subsystem

#### PEAKtable:DATA:ALL?

Queries peaktable with frequency and level values for multiple traces. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions (see DATA:ALL? commands).

#### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
FreqID	Enum	Optional: FREQUENCY
FreqElements	ULong	Number of FreqID Elements
FreqList	FlexArray[Double]	List of frequency values in Hz
TransID	Enum	Optional: TRANSMITTER
TransElements	ULong	Number of TransID Elements
TransList	FlexArray[String]	List of transmitter names
TraceID	Enum	Optional: RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS
TraceElements	ULong	Number of TraceID Elements
Overdriven	Bool	Overdriven flag
NotRealtime	Bool	NotRealtime flag
LevelList	FlexArray[Float]	List of power levels in actual unit

#### Examples

//Queries all activated traces of the peaktable

PEAKtable:DATA:ALL?

0,1551350855,82909033,9242,FREQUENCY,2,104815669.821,97703112.7976,RMS,4,0,0,-57.17,-57.71

#### PEAKtable:DATA:COUNT?

Queries the number of peaktable list elements

## Result Parameter

Parameter	Type	Description
Value	ULong	Number of peaktable list elements

## Examples

//Queries the number of peaktable list elements

PEAK:DATA:COUN?

6

## PEAKtable:DATA:FREQuency?

Queries peaktable frequency data

## Result Parameter

Parameter	Type	Description
Data	FlexArray[Double]	List of peaktable frequencies in Hz

## Examples

//Queries the peaktable frequencies

PEAK:DATA:FREQ?

94699563.7042,90104272.4717,90096905.6252,92211000.0235,93499880.9218,91995584.0599

## PEAKtable:DATA:LEVel?

Queries peaktable level data

## Query Parameter

Parameter	Type	Description
Name	Enum	RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

## Result Parameter

Parameter	Type	Description
Data	FlexArray[Float]	List of power levels in actual unit

## Examples

//Queries the peaktable power levels for the RMS trace

PEAK:DATA:LEV? RMS

-112.30,-112.51,-113.16,-113.44,-117.04,-119.30

## PEAKtable:DATA:SCAN:NUMBer?

PEAKtable data scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

### Examples

Queries the data scan number

PEAKtable:DATA:SCAN:NUMBER?

1624

Queries the data scan number

PEAK:DATA:SCAN:NUMB?

7086

### PEAKtable:DATA:TSTamp?

Query PEAKtable data timestamp

### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

### Examples

Queries the PEAKtable data timestamp

PEAKtable:DATA:TSTamp?

0,1564392802,584232802

Queries the PEAKtable data timestamp

PEAK:DATA:TST?

0,1564392812,124232802

### PEAKtable:DATA:UPDate?

Update is available in RUN and HOLD mode (see DATA:UPDate Commands)

### Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong, Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

### Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.



## Examples

//Peaktable data update with old scancount = 10 and timeout = 100ms

PEAK:DATA:UPD? 10,100ms

16088

//Force peaktable data update because we do not have old scan number

PEAKtable:DATA:UPDate?

15921

//Peaktable data update with old scan number = 0

PEAKtable:DATA:UPDate? 0

15975

//Force peaktable data update with timeout = 100ms

PEAKtable:DATA:UPDate? NaN,100ms

16031

## PEAKtable:SCAN:NUMBer?

Queries the peaktable scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

## Examples

//Queries the current scan number

PEAK:SCAN:NUMB?

2087

## PEAKtable:TRACe

Selects the active trace for the peaktable. Only traces enabled by SPECTrum:TRACe:ENABLE are available.

### Command Parameter

Parameter	Type	Description
Name	Enum	RMS, PPK, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

## Examples

//Sets AvR as active trace for peaktable

PEAK:TRAC AvR

## PEAKtable:TRACe?

Queries the active trace

### Result Parameter

Parameter	Type	Description
Name	Enum	RMS, PPk, MPk, Avg, Smp, MnR, AvR, MxR, MxP, MnP, MxA, MxS

### Examples

//Queries the current active trace for peaktable

PEAK:TRAC?

RMS

### PEAKtable:TSTamp?

Queries peaktable data timestamp

### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

### Examples

//Queries the peaktable data timestamp

PEAK:TST?

0,1533062041,302777397

## Narda Command Reference - STream

Stream subsystem

### STream:ADD?

Adds a new stream to the active stream task

#### Query Parameter

Parameter	Type	Description
Type	String	Stream type e.g. 'VITA49_IQ', 'VITA49_SPECTRUM_RMS', 'VITA49_SPECTRUM_PLUS_P K', 'VITA49_SPECTRUM_THIRD_DETECTOR', 'AUDIO_DEMOD'

#### Result Parameter

Parameter	Type	Description
Index	UShort	Index of the new stream added

#### Examples

```
//Adds a new 'VITA49_IQ' stream
```

```
STR:ADD? 'VITA49_IQ'
```

```
1
```

### STream:CONNection:ADDRess

Sets the connection IP or multicast address for selected stream

#### Command Parameter

Parameter	Type	Description
Value	String	Stream IP or Multicast Address for the socket connection

#### Examples

```
//Enables localhost loopback with TCP fast path option SIO_LOOPBACK_FAST_PATH
```

```
STR:CONN:ADDR 'LoopbackFastPath'
```

```
//Sets IPv4 address
```

```
STR:CONN:ADDR 192.168.128.1
```

```
//Sets IPv4 multicast address
```

```
STR:CONN:ADDR 226.0.0.1
```

```
//Sets IPv4 localhost loopback
```

```
STR:CONN:ADDR '127.0.0.1'
```

```
//Sets localhost loopback
```

```
STR:CONN:ADDR 'localhost'
```

### STReam:CONNection:ADDReSS?

Queries the stream IP address

#### Result Parameter

Parameter	Type	Description
Value	String	Stream IP or Multicast Address for the socket connection

#### Examples

//Queries the address

STR:CONN:ADDR?

"192.168.128.1"

### STReam:CONNection:CLOSe

Closes connection for selected stream

#### Examples

//Closes connection

STR:CONN:CLOS

### STReam:CONNection:IDN

Stream identifier for the selected stream connection

#### Command Parameter

Parameter	Type	Description
Id	ULong	Id of the stream

#### Examples

//Sets identifier 12345 to the stream connection

STR:CONN:IDN 12345

### STReam:CONNection:IDN?

Queries the stream identifier of the stream connection

#### Result Parameter

Parameter	Type	Description
Id	ULong	Id of the stream

#### Examples

//Queries the identifier of the stream connection

STR:CONN:IDN?

0

## STReam:CONNection:OPEN

Opens connection for selected stream

### Examples

//Opens connection

STR:CONN:OPEN

## STReam:CONNection:PORT

Connection portnumber for selected stream

### Command Parameter

Parameter	Type	Description
Value	UShort	Portnumber for the socket connection

### Examples

//Sets socket portnumber 55555

STR:CONN:PORT 55555

## STReam:CONNection:PORT?

Queries the connection portnumber for selected stream

### Result Parameter

Parameter	Type	Description
Value	UShort	Portnumber for the socket connection

### Examples

//Queries socket portnumber

STR:CONN:PORT?

4444

## STReam:CONNection:STATe?

Queries connection state for selected stream

### Result Parameter

Parameter	Type	Description
Value	Enum	Connection state (OPENED, CLOSED, CONNECTED)

### Examples

//Queries connection state

STR:CONN:STAT?

CLOSED

### STReam:CONNection:TYPE

Sets the connection type TCP or UDP for selected stream

#### Command Parameter

Parameter	Type	Description
Value	Enum	Connection type (TCP_SERVER, TCP_CLIENT, UDP_SINGLECAST, UDP_MULTICAST)

#### Examples

//Sets connection type to UDP

STR:CONN:TYPE UDP\_SINGLECAST

### STReam:CONNection:TYPE?

Queries the connection type for selected stream

#### Result Parameter

Parameter	Type	Description
Value	Enum	Connection type

#### Examples

//Queries the connection type

STR:CONN:TYPE?

UDP\_SINGLECAST

### STReam:DELeTe

Deletes a stream from the active stream task

#### Command Parameter

Parameter	Type	Description
Index	UShort	Index of the stream to delete

#### Examples

//Deletes the stream with the index 1

STR:DEL 1

### STReam:LIST?

Queries a list of streams from the active stream task

**Result Parameter**

Parameter	Type	Description
List	FlexArray[String,UShort]	List of available streams in currently active task (Type, Index)

**Examples**

//Queries a list of streams in the currently active RT Spectrum task

STR:LIST?

"AUDIO\_DEMOD",1

//Queries a list of streams in the currently active RT Streaming task

STR:LIST?

"VITA49\_IQ",1

//Queries a list of streams in the currently active RT Streaming task

STR:LIST?

"VITA49\_SPECTRUM\_RMS",2,"VITA49\_IQ",1,"VITA49\_SPECTRUM\_PLUS\_PK",3,"VITA49\_SPECTRUM\_THIRD\_DETECTOR",4

**STReam:SElect**

Selects a stream of the active stream task for configuration of the connection

**Command Parameter**

Parameter	Type	Description
Index	UShort	Index of the stream to select

**Examples**

//Selects the stream with the index 2 of the active stream task

STR:SEL 2

**STReam:SElect?**

Queries the selected stream of the active stream task

**Result Parameter**

Parameter	Type	Description
Index	UShort	Index of the selected stream

**Examples**

//Queries the selected stream of the active stream task

STR:SEL?

2

## Narda Command Reference - IQStream

IQ Stream subsystem

### IQStream:ANTenna:FACTor?

Queries the antenna factor for Ftune

#### Result Parameter

Parameter	Type	Description
Value	Double	Antenna Factor for FTune

#### Examples

//Queries the antenna factor for Ftune

IQST:ANT:FACT?

50.3224864799

### IQStream:CBW

Sets the IQ channel bandwidth

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth (Hz, kHz, MHz, GHz)

#### Examples

//Sets the bandwidth in kHz

IQST:CBW 100 kHz

### IQStream:CBW?

Queries the IQ channel bandwidth

#### Result Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth in Hz

#### Examples

//Queries the bandwidth

IQST:CBW?

100000

### IQStream:CBW:LIST?

Queries a list of possible CBW settings



**Result Parameter**

Parameter	Type	Description
List	FlexArray[Double]	List of CBWs

**Examples**

//Queries a list of CBWs

IQST:CBW:LIST?

2000000,1920000,1600000,1500000,1280000,1250000,1200000,1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000,300000,256000,250000,240000,200000,192000,160000,150000,128000,125000,120000,100000,960000,800000,750000,64000,62500,60000,51200,50000,48000,40000,38400,32000,30000,25600,25000,24000,20000,19200,16000,15000,12800,12500,12000,10000,9600,9000,8000,7500,6400,6250,6000,5120,5000,4800,4000,3840,3200,3100,3000,2700,2560,2500,2400,2100,2000,1920,1600,1500,1280,1250,1200,1000,960,800,750,640,625,600,512,500,480,400,384,320,300,256,250,240,200,192,160,150,128,125,120,100,50,25

**IQStream:FREQuency:TUNE**

Sets the IQ tune frequency

**Command Parameter**

Parameter	Type	Description
Value	Frequency	Tune frequency (Hz, kHz, MHz, GHz)

**Examples**

//Sets the tune frequency in MHz

IQST:FREQ:TUNE 1.2 MHz

**IQStream:FREQuency:TUNE?**

Queries the IQ tune frequency

**Result Parameter**

Parameter	Type	Description
Value	Frequency	Tune frequency in Hz

**Examples**

//Queries the tune frequency

IQST:FREQ:TUNE?

1200000

**IQStream:OVERsampling**

Sets the IQ oversampling state

### Command Parameter

Parameter	Type	Description
Value	Bool	Oversampling

### Examples

//Disables oversampling

IQST:OVER OFF

### IQStream:OVERsampling?

Queries the IQ oversampling state

### Result Parameter

Parameter	Type	Description
Value	Bool	Oversampling

### Examples

//Queries the oversampling state

IQST:OVER?

0

### IQStream:PAYLoad:FORMat

Sets the IQ payload format

### Command Parameter

Parameter	Type	Description
Value	Enum	IQ payload format (I16Q16, I32Q32)

### Examples

//Sets the IQ format to 16 bit

IQST:PAYL:FORM I16Q16

### IQStream:PAYLoad:FORMat?

Queries the IQ payload format

### Result Parameter

Parameter	Type	Description
Value	Enum	IQ payload format

### Examples

//Queries the IQ format

IQST:PAYL:FORM?

I16Q16

## IQSTream:PAYLoad:LIMit

Sets the IQ payload limit in words (32Bit)

### Command Parameter

Parameter	Type	Description
Value	ULong	Payload Limit (256,512,1024,2048)

### Examples

//Sets the payload limit to 2048 words

IQST:PAYL:LIM 2048

## IQSTream:PAYLoad:LIMit?

Queries the IQ payload limit

### Result Parameter

Parameter	Type	Description
Value	ULong	Payload Limit

### Examples

//Queries the payload limit

IQST:PAYL:LIM?

2048

## IQSTream:PAYLoad:LIMit:LIST?

Queries a list of possible payload limits depending on payload format, CBW and oversampling

### Result Parameter

Parameter	Type	Description
Value	FlexArray[ULong]	List of possible payload limits

### Examples

//Queries list of possible limits

IQST:PAYL:LIM:LIST?

4,8,16,32,64,128,256,512,1024,2048

### Narda Command Reference - BEARing

Bearing subsystem

#### BEARing:AZIMuth:CORRection

Sets the azimuth correction for bearing

##### Command Parameter

Parameter	Type	Description
Value	Double	Azimuth correction in degree

##### Examples

//Sets the azimuth correction in degree

BEAR:AZIM:CORR 1.2

#### BEARing:AZIMuth:CORRection?

Queries the azimuth correction for bearing

##### Result Parameter

Parameter	Type	Description
Value	Double	Azimuth correction in degree

##### Examples

//Queries the azimuth correction

BEAR:AZIM:CORR?

1.2

#### BEARing:CBW

Sets the channel bandwidth for bearing

##### Command Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth (Hz, kHz, MHz, GHz)

##### Examples

//Sets the bandwidth in kHz

BEAR:CBW 100 kHz

#### BEARing:CBW?

Queries the channel bandwidth for bearing

**Result Parameter**

Parameter	Type	Description
Value	Frequency	Channel Bandwidth in Hz

**Examples**

//Queries the bandwidth

BEAR:CBW?

100000

**BEARing:CBW:LIST?**

Queries a list of possible CBW settings

**Result Parameter**

Parameter	Type	Description
List	FlexArray[Double]	List of CBWs

**Examples**

//Queries a list of CBWs

BEAR:CBW:LIST?

40000000,38400000,32000000,30000000,25600000,25000000,24000000,20000000,19200000,16000000,15000000,12800000,12500000,12000000,10000000,9600000,8000000,7500000,6400000,6250000,6000000,5120000,5000000,4800000,4000000,3840000,3200000,3000000,2560000,2500000,2400000,2000000,1920000,1600000,1500000,1280000,1250000,1200000,1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000,300000,256000,250000,240000,200000,192000,160000,150000,128000,125000,120000,100000,96000,80000,75000,64000,62500,60000,51200,50000,48000,40000,38400,32000,30000,25600,25000,24000,20000,19200,16000,15000,12800,12500,12000,10000,9600,9000,8000,7500,6400,6250,6000,5120,5000,4800,4000,3840,3200,3100,3000,2700,2560,2500,2400,2100,2000,1920,1600,1500,1280,1250,1200,1000,960,800,750,640,625,600,512,500,480,400,384,320,300,256,250,240,200,192,160,150,128,125,120,100,50,25

**BEARing:CYCLe:TIME?**

Queries the cycle time for bearing

**Result Parameter**

Parameter	Type	Description
Value	Timespan	Cycle time in s

**Examples**

//Queries cycle time

BEAR:CYCL:TIME?

0.0434810703125

**BEARing:DATA:ALL?**

Queries all bearing values. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)

## Command Reference Guide

### Command Descriptions

#### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
BearingID	Enum	Optional: BEARING
BearingElements	ULong	Number of BearingID Elements
Overdriven	Bool	Overdriven flag
VectorLost	Bool	Vectors Lost flag
ValidBearing	Bool	Valid Bearing flag
Azimuth	Float	Azimuth in degree.
AzimuthCorrection	Double	Azimuth correction in degree.
Elevation	Float	Elevation in degree.
DFQuality	Float	DFQuality in percent.
DetectorValue	Float	Omnidirectional Detector value
CompassID	Enum	Optional: COMPASS
CompassElements	ULong	Number of CompassID Elements
CompassAzimuth	Float	Azimuth in degree.
CompassElevation	Float	Elevation in degree.
CompassRoll	Float	Roll in degree.
GnssID	Enum	Optional: GNSS
GnssElements	ULong	Number of GnssID Elements
GnssFrozenFlag	Bool	Is true if GNSS signal is frozen.
Gnss3DFlag	Bool	Is true if GNSS signal is 3D.
GnssSatellites	UShort	Number of Satellites
GnssLatitude	Double	Latitude
GnssLongitude	Double	Longitude
GnssAltitude	Float	Altitude in m
GnssSpeed	Float	Speed in m/s
GnssCourse	Float	Course in degree

#### Examples

//Queries all bearing values (with GNSS reception)

BEAR:DATA:ALL?

0,1551352848,319361484,1161,BEARING,8,0,0,1,291.189,0,20.166,58.741,-  
103.56,COMPASS,3,235,2.6,0,GNSS,8,0,1,5,48.45829,9.23033,401.6,0.041,354.29

//Queries all bearing values (without GNSS reception)

BEAR:DATA:ALL?

0,1551351255,94004555,119,BEARING,8,0,0,1,50.837,0,-7.014,50.824,-98.85,COMPASS,3,235.1,2.8,0

## BEARing:DATA:AZIMuth?

Queries the azimuth of the bearing result.

### Result Parameter

Parameter	Type	Description
Value	Float	Azimuth of bearing in degree

### Examples

//Queries the azimuth (in degree) of the last bearing result

BEAR:DATA:AZIM?

211.2865

## BEARing:DATA:DETECTOR?

Queries the bearing detector value

### Result Parameter

Parameter	Type	Description
Value	Float	Omnidirectional Detector value

### Examples

//Queries the bearing detector value

BEAR:DATA:DET?

-16.79

## BEARing:DATA:DFQuality?

Queries the DF quality of the current bearing in percent

### Result Parameter

Parameter	Type	Description
Value	Float	Quality of the bearing

### Examples

//Queries the DF quality of the last bearing in percent

BEAR:DATA:DFQ?

87.1462

## BEARing:DATA:ELEVation?

Queries the elevation of the bearing result.

### Result Parameter

Parameter	Type	Description
Value	Float	Elevation of bearing in degree.

### Examples

//Queries the elevation (in degree) of the last bearing.

BEAR:DATA:ELEV?

37.5

### BEARing:DATA:SCAN:NUMBer?

Queries the bearing data scan number

#### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

### Examples

//Queries the data scan number

BEARing:DATA:SCAN:NUMBer?

5072

//Queries the data scan number

BEAR:DATA:SCAN:NUMB?

5114

### BEARing:DATA:TSTamp?

Queries the bearing data timestamp

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

### Examples

//Queries the BEARing data timestamp

BEARing:DATA:TSTamp?

0,1564485640,97158964

//Queries the BEARing data timestamp

BEAR:DATA:TST?

0,1564485649,740908964

### BEARing:DATA:UPDate?

Update is available in RUN and HOLD mode (see DATA:UPDate Commands)



## Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong, Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

## Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.

## Examples

//Bearing data update with old scancount = 10 and timeout = 100ms

BEAR:DATA:UPD? 10,100ms

57182

//Force Bearing data update because we do not have old scan number

BEAR:DATA:UPD?

57026

//Bearing data update with old scan number = 0

BEAR:DATA:UPD? 0

57125

//Force Bearing data update with timeout = 100ms

BEAR:DATA:UPD? NaN,100ms

57153

## BEARing:DATA:VALid?

Queries the valid flag of the last bearing result

## Result Parameter

Parameter	Type	Description
Value	Bool	Valid flag of bearing data.

## Examples

//Queries the valid flag of the last bearing result

BEAR:DATA:VAL?

1

## BEARing:DATA:VALid:ENABLE

Switches 'Last valid bearing data' flag on/off. When enabled, last valid bearing is returned and not the last measurement run data. Filter settings like 'BEARing:DFSquelch', 'BEARing:MIN:DFQuality' or 'BEARing:MIN:STABILITY' will only take effect, if this parameter is enabled!

#### Command Parameter

Parameter	Type	Description
Value	Bool	Last valid bearing data

#### Examples

//Enables valid bearing data only

BEAR:DATA:VAL:ENAB 1

### BEARing:DATA:VALid:ENABLE?

Queries last valid bearing data state. When enabled, last valid bearing is returned and not the last measurement run data. Filter settings like 'BEARing:DFSquelch', 'BEARing:MIN:DFQuality' or 'BEARing:MIN:STABility' will only take effect, if this parameter is enabled!

#### Result Parameter

Parameter	Type	Description
Value	Bool	Last valid bearing data

#### Examples

//Queries last valid bearing data state

BEAR:DATA:VAL:ENAB?

1

### BEARing:DFCorrection:LIST?

Queries a list of DF correction files

#### Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of DF correction files

#### Examples

//Queries a list of DF correction files

BEAR:DFC:LIST?

"None","DF-COR-01\_3361\_01\_PT-1005","DF-COR-01\_3361\_01\_\_2items","DF-COR-01\_3361\_01\_\_3items"

### BEARing:DFCorrection:SElect

Selects the DF correction file. A DFCorrection file must be uploaded to the SignalShark and saved under D:Narda\_SignalShark\Configuration\DFCorrection. DFCorrFile is the name of the .xml-file thus for MyFile.xml is the DFCorrFile parameter "MyFile".

Note: It is only possible to select or query a DF correction file when an Auto DF-task is selected (see TASK:SElect and TASK:SElect? commandos to select a task and query the selected task)

#### Command Parameter

Parameter	Type	Description
DFCorrFile	String	Name of the DF correction file

## Examples

//Selects the file 'MyFile' for DF correction

BEARing:DFCorrection:SElect 'MyFile'

## BEARing:DFCorrection:SElect?

Queries the selected DF correction file. A DFCorrection file must be uploaded to the SignalShark and saved under D:Narda\_SignalShark\Configuration\DFCorrection. DFCorrFile is the name of the .xml-file thus for MyFile.xml is the DFCorrFile parameter "MyFile".

Note: It is only possible to select or query a DF correction file when an Auto DF-task is selected (see TASK:SElect and TASK:SElect? commandos to select a task and query the selected task)

## Result Parameter

Parameter	Type	Description
DFCorrFile	String	Name of the DF correction file

## Examples

//Queries the selected file for DF correction

BEARing:DFCorrection:SElect?

"DF-COR-01\_3361\_01\_\_3items"

## BEARing:DFSquelch

Sets the DF squelch for bearing. The flags BEARing:DFSquelch:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:DFSquelch to take effect!

## Command Parameter

Parameter	Type	Description
Value	Unit	DF Squelch level in actual unit or units of: No antenna connected: dBm, dBV, dBmV, dBuV  With antenna connected: dBm, dBV, dBmV, dBuV, W_m2, W_cm2, A_m, V_m, dBA_m, dBV_m, dBmV_m, dBuV_m

## Examples

//Sets the DF squelch to -80 dBm

BEAR:DFSQ -80 dBm

## BEARing:DFSquelch?

Queries the DF squelch level for bearing. The flags BEARing:DFSquelch:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:DFSquelch to take effect!

#### Result Parameter

Parameter	Type	Description
Value	Unit	DF Squelch level in actual unit

#### Examples

//Queries the DF squelch in actual unit

BEAR:DFSQ?

-90.00

### BEARing:DFSQuelch:ENABLE

Enables/Disables DF squelch criteria for bearings. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:DFSQuelch:ENABLE to take effect.

#### Command Parameter

Parameter	Type	Description
Value	Bool	DF Squelch state

#### Examples

//Enables DF squelch criteria

BEAR:DFSQ:ENAB ON

### BEARing:DFSQuelch:ENABLE?

Queries the DF squelch state. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:DFSQuelch:ENABLE to take effect.

#### Result Parameter

Parameter	Type	Description
Value	Bool	DF Squelch state

#### Examples

//Queries the DF squelch state

BEAR:DFSQ:ENAB?

1

### BEARing:FREQuency:TUNE

Sets the bearing tune frequency

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency (Hz, kHz, MHz, GHz)

## Examples

//Sets the tune frequency in MHz

BEAR:FREQ:TUNE 1.2 MHz

## BEARing:FREQuency:TUNE?

Queries the bearing tune frequency

### Result Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency in Hz

## Examples

//Queries the tune frequency

BEAR:FREQ:TUNE?

446000000

## BEARing:FREQuency:TUNE:STEP

Sets the tune frequency step for bearing

### Command Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency step (Hz, kHz, MHz, GHz)

## Examples

//Sets the tune frequency step in MHz

BEAR:FREQ:TUNE:STEP 1 MHz

## BEARing:FREQuency:TUNE:STEP?

Queries the tune frequency step for bearing

### Result Parameter

Parameter	Type	Description
Value	Frequency	Tune frequency step in Hz

## Examples

//Queries the tune frequency step

BEAR:FREQ:TUNE:STEP?

1000000

## BEARing:MEASurement:TIME

Sets the measurement time for bearing

#### Command Parameter

Parameter	Type	Description
Value	Timespan	Measurement time (h, m, s, ms)

#### Examples

//Sets the measurement time to 10ms

BEAR:MEAS:TIME 10 ms

#### BEARing:MEASurement:TIME?

Queries the measurement time for bearing

#### Result Parameter

Parameter	Type	Description
Value	Timespan	Measurement time in s

#### Examples

//Queries the measurement time

BEAR:MEAS:TIME?

0.001

#### BEARing:MIN:DFQuality

Sets the minimum DF quality for a bearing. The flags BEARing:MIN:DFQuality:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:MIN:DFQuality to take effect!

#### Command Parameter

Parameter	Type	Description
DFMin	Float	minimum DF quality

#### Examples

//Sets the min DF quality to 80 percent.

BEARing:MIN:DFQuality 80

#### BEARing:MIN:DFQuality?

Queries the minimum DF quality for a bearing. Sets the minimum DF quality for a bearing. The flags BEARing:MIN:DFQuality:ENABLE and BEARing:DATA:VALid:ENABLE must be enabled for the settings of BEARing:MIN:DFQuality to take effect!

#### Result Parameter

Parameter	Type	Description
DFMin	Float	minimum DF quality

## Examples

```
//Queries the min DF quality
```

```
BEARing:MIN:DFQuality?
```

```
60
```

## BEARing:MIN:DFQuality:ENABLE

Enables/Disables minimum DF quality criteria for bearings. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:DFQuality:ENABLE to take effect

### Command Parameter

Parameter	Type	Description
Value	Bool	min DF quality state

## Examples

```
//Enables min DF quality criteria
```

```
BEAR:MIN:DFQ:ENAB ON
```

## BEARing:MIN:DFQuality:ENABLE?

Queries the minimum DF quality criteria state. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:DFQuality:ENABLE to take effect.

### Result Parameter

Parameter	Type	Description
Value	Bool	min DF quality state

## Examples

```
//Queries min DF quality criteria state
```

```
BEAR:MIN:DFQ:ENAB?
```

```
1
```

## BEARing:MIN:STABILITY

Sets the minimum stability for bearings. The flags BEARing:DATA:VALid:ENABLE and BEARing:MIN:STABILITY:ENABLE must be enabled for the settings of BEARing:MIN:STABILITY to take effect!

### Command Parameter

Parameter	Type	Description
Value	Unit	Min Stability in (dB)

## Examples

```
//Sets the min stability to 10 dB
```

```
BEAR:MIN:STAB 10
```

### BEARing:MIN:STABility?

Queries the minimum level stability for bearings. The flags BEARing:DATA:VALid:ENABLE and BEARing:MIN:STABility:ENABLE must be enabled for the settings of BEARing:MIN:STABility to take effect!

#### Result Parameter

Parameter	Type	Description
Value	Unit	Min Stability in (dB)

#### Examples

//Queries the min stability

BEAR:MIN:STAB?

10

### BEARing:MIN:STABility:ENABLE

Enables/Disables minimum stability criteria for bearings. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:STABility:ENABLE to take effect.

#### Command Parameter

Parameter	Type	Description
Value	Bool	min stability state

#### Examples

//Enables min stability criteria

BEAR:MIN:STAB:ENAB ON

### BEARing:MIN:STABility:ENABLE?

Queries the minimum stability criteria state. The flag BEARing:DATA:VALid:ENABLE must also be enabled for the settings of BEARing:MIN:STABility:ENABLE to take effect.

#### Result Parameter

Parameter	Type	Description
Value	Bool	min stability state

#### Examples

//Queries min stability criteria state

BEAR:MIN:STAB:ENAB?

0

### BEARing:NORTH:REFerence

Sets the north reference for bearing



### Command Parameter

Parameter	Type	Description
Value	Enum	Type (COMPASS, GNSS_VELOCITY, REFERENCE_MARK_DIRECTION)

### Examples

```
//Sets the north reference
```

```
BEAR:NORT:REF COMPASS
```

## BEARing:NORTH:REfERENCE?

Queries the north reference for bearing

### Result Parameter

Parameter	Type	Description
Value	Enum	Type (COMPASS, GNSS_VELOCITY, REFERENCE_MARK_DIRECTION)

### Examples

```
//Queries the north reference
```

```
BEAR:NORT:REF?
```

```
REFERENCE_MARK_DIRECTION
```

## BEARing:POST:AVG

Sets the post averaging time for bearing

### Command Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time (h, m, s, ms)

### Examples

```
//Sets the post averaging time to 10ms
```

```
BEAR:POST:AVG 0.01
```

## BEARing:POST:AVG?

Queries the post averaging time for bearing

### Result Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time in s

### Examples

//Queries the post averaging time

BEAR:POST:AVG?

0

## BEARing:REfERENCE:MARK:DIRection

Sets the reference mark direction for bearing

### Command Parameter

Parameter	Type	Description
Value	Double	Reference mark direction in degrees

### Examples

//Sets the reference mark direction in degree

BEAR:REF:MARK:DIR 1.2

## BEARing:REfERENCE:MARK:DIRection?

Queries the reference mark direction for bearing

### Result Parameter

Parameter	Type	Description
Value	Double	Reference mark direction in degrees

### Examples

//Queries the reference mark direction

BEAR:REF:MARK:DIR?

0

## BEARing:SCAN:COUNt

Sets the bearing scan count

### Command Parameter

Parameter	Type	Description
Value	ULong	Scan count

### Examples

//Sets the scan count to 50

BEAR:SCAN:COUN 50

## BEARing:SCAN:COUNt?

Queries the bearing scan count

### Result Parameter

Parameter	Type	Description
Value	ULong	Scan count

### Examples

//Queries the scan count

BEAR:SCAN:COUN?

50

## BEARing:SCAN:NUMBer?

Queries the bearing scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

### Examples

//Queries the current scan number

BEAR:SCAN:NUMB?

46301

## BEARing:SCAN:TIME?

Queries the scan time for bearing

### Result Parameter

Parameter	Type	Description
Value	Timespan	Scan time in s

### Examples

//Queries the scan time

BEAR:SCAN:TIME?

0.0535433359375

## BEARing:TSTamp?

Queries the bearing data timestamp

### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part

### Examples

//Queries the bearing data timestamp

BEAR:TST?

0,1550654793,460288647

## Narda Command Reference - DEMod

Demodulation subsystem

### DEMod:CBW

Sets the channel bandwidth for demodulation

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth (Hz, kHz, MHz, GHz)

#### Examples

```
//Sets the bandwidth in kHz
```

```
DEMod:CBW 100 kHz
```

### DEMod:CBW?

Queries the channel bandwidth for demodulation

#### Result Parameter

Parameter	Type	Description
Value	Frequency	Channel Bandwidth in Hz

#### Examples

```
//Queries the bandwidth
```

```
DEMod:CBW?
```

```
100000
```

### DEMod:CBW:FILTer:TYPE

Sets the demodulation CBW filter type

#### Command Parameter

Parameter	Type	Description
Value	Enum	Filter type (Channel, MIL, CISPR)

#### Examples

```
//Sets the cbw filter type to Channel
```

```
DEMod:CBW:FILT:TYPE Channel
```

### DEMod:CBW:FILTer:TYPE?

Queries the demodulation CBW filter type

### Result Parameter

Parameter	Type	Description
Value	Enum	Filter type (Channel, MIL, CISPR)

### Examples

//Queries the cbw filter type

DEM:CBW:FILT:TYPE?

CHANNEL

### DEMod:CBW:LIST?

Queries a list of possible CBW settings

### Result Parameter

Parameter	Type	Description
List	FlexArray[Double]	List of CBWs

### Examples

//Queries a list of CBWs

DEM:CBW:LIST?

1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000,300000,256000,250000,240000,200000,192000,160000,150000,128000,125000,120000,100000,96000,80000,75000,64000,62500,60000,51200,50000,48000,40000,38400,32000,30000,25600,25000,24000,20000,19200,16000,15000,12800,12500,12000,10000,9600,9000,8000,7500,6400,6250,6000,5120,5000,4800,4000,3840,3200,3100,3000,2700,2560,2500,2400,2100,2000,1920,1600,1500,1280,1250,1200,1000,960,800,750,640,625,600,512,500,480,400,384,320,300,256,250,240,200,192,160,150,128,125,120,100,50,25

### DEMod:CBW:OVERsampling

Sets the demodulation oversampling state

### Command Parameter

Parameter	Type	Description
Value	Bool	Oversampling

### Examples

//Disables oversampling

DEM:CBW:OVER OFF

### DEMod:CBW:OVERsampling?

Queries the demodulation oversampling state

### Result Parameter

Parameter	Type	Description
Value	Bool	Oversampling

## Examples

//Queries the oversampling state

DEM:CBW:OVER?

0

## DEMod:FREQuency:TUNE

Sets the demodulation tune frequency

### Command Parameter

Parameter	Type	Description
Value	Frequency	Demod tune frequency in Hz

## Examples

//Sets the demodulation tune frequency

DEM:FREQ:TUNE 97.7 MHz

## DEMod:FREQuency:TUNE?

Queries the demodulation tune frequency

### Result Parameter

Parameter	Type	Description
Value	Frequency	Demod tune frequency in Hz

## Examples

//Queries the demodulation tune frequency

DEM:FREQ:TUNE?

97700000

## DEMod:FREQuency:TUNE:AFC

Switches the demodulation AFC on/off

### Command Parameter

Parameter	Type	Description
Value	Bool	AFC Enable Value

## Examples

//Enables AFC

DEM:FREQ:TUNE:AFC ON

## DEMod:FREQuency:TUNE:AFC?

Queries the demodulation AFC state

### Result Parameter

Parameter	Type	Description
Value	Bool	AFC Enable Value

### Examples

//Queries the AFC state

DEM:FREQ:TUNE:AFC?

1

## DEMod:FREQuency:TUNE:BFO

Sets the demodulation BFO value (only for USB,ISB,LSB)

### Command Parameter

Parameter	Type	Description
Value	Frequency	BFO Value in Hz

### Examples

//Sets the BFO value

DEM:FREQ:TUNE:BFO 100

## DEMod:FREQuency:TUNE:BFO?

Queries the demodulation BFO value (only for USB,ISB,LSB)

### Result Parameter

Parameter	Type	Description
Value	Frequency	BFO Value in Hz

### Examples

//Queries the BFO value

DEM:FREQ:TUNE:BFO?

100

## DEMod:MUTE

Sets the mute state of the demodulation volume

### Command Parameter

Parameter	Type	Description
Value	Bool	Demod volume mute state ON/OFF

### Examples

//Enables muting for the demodulations

DEM:MUTE ON



## DEMod:MUTE?

Queries the mute state of the demodulation volume

### Result Parameter

Parameter	Type	Description
Value	Bool	Demod volume mute state

### Examples

//Queries the mute state of the demodulation volume

DEMod:MUTE?

1

## DEMod:SQUelch

Sets the demodulation squelch value. The command DEMod:SQUelch:ENABLE must be enabled for the settings of DEMod:SQUelch to take effect!

### Command Parameter

Parameter	Type	Description
Value	Unit	SQUelch Value (dBm)

### Examples

//Sets the squelch value to 10 dBm

DEMod:SQU 10 dBm

## DEMod:SQUelch?

Queries the demodulation squelch value. The command DEMod:SQUelch:ENABLE must be enabled for the settings of DEMod:SQUelch to take effect!

### Result Parameter

Parameter	Type	Description
Value	Unit	SQUelch Value

### Examples

//Queries the squelch value

DEMod:SQU?

10.00

## DEMod:SQUelch:ENABLE

Switches the demodulation squelch on/off

### Command Parameter

Parameter	Type	Description
Value	Bool	SQUelch Enable Value

### Examples

//Sets the squelch enable value

DEM:SQU:ENAB 1

### DEMod:SQUelch:ENABLE?

Queries the demodulation squelch enable state

#### Result Parameter

Parameter	Type	Description
Value	Bool	SQUelch Enable Value

### Examples

//Queries the squelch enable value

DEM:SQU:ENAB?

1

### DEMod:TYPE

Sets the demodulation type

#### Command Parameter

Parameter	Type	Description
Value	Enum	Demod type (OFF, FM, PM, AM, AM_FM, PULSE, CW, ISB, LSB, USB, IQ)

### Examples

//Sets the demodulation type to AM

DEM:TYPE AM

### DEMod:TYPE?

Queries the demodulation type

#### Result Parameter

Parameter	Type	Description
Value	Enum	Demod type

### Examples

//Queries the demodulation type

DEM:TYPE?

AM

### DEMod:TYPE:ENABLE

Switches the demodulation type on/off

## Command Parameter

Parameter	Type	Description
Value	Bool	Demod Enable Value

## Examples

//Sets the demod enable value

DEM:TYPE:ENAB 1

## DEMod:TYPE:ENABLE?

Queries the demodulation type enable state

## Result Parameter

Parameter	Type	Description
Value	Bool	Demod Enable Value

## Examples

//Queries the demod enable value

DEM:TYPE:ENAB?

1

## DEMod:VOLume

Sets the demodulation volume

## Command Parameter

Parameter	Type	Description
Value	Double	Demod volume 0%...100%

## Examples

//Sets the demodulation volume to 90%

DEM:VOL 90

## DEMod:VOLume?

Queries the demodulation volume

## Result Parameter

Parameter	Type	Description
Value	Double	Demod volume

## Examples

//Queries the demodulation volume

DEM:VOL?

90

### DEMod:VOLume:AGC

Switches the demodulation AGC on/off

#### Command Parameter

Parameter	Type	Description
Value	Bool	AGC Enable Value

#### Examples

//Enables AGC

DEMod:VOL:AGC ON

### DEMod:VOLume:AGC?

Queries the demodulation AGC state

#### Result Parameter

Parameter	Type	Description
Value	Bool	AGC Enable Value

#### Examples

//Queries the AGC state

DEMod:VOL:AGC?

1

## Narda Command Reference - SGRam

Spectrogram subsystem

### SGRam:DATA:ALL?

Queries all or a range of spectrogram frames. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)

#### Query Parameter

Parameter	Type	Description
Optional	Optional[ULongEnum]	First Frame to Query or MIN, Last Frame to Query or MAX, RMS, PPk, MPk, Avg, Smp

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
InfoID	Enum	Optional: INFO
InfoElements	ULong	Number of InfoID Elements
FramesQueried	ULong	Number of Frames queried
FramesAvaliable	ULong	Number of Frames available
TotalBins	ULong	Number of Bins per Frame
FreqStart	Frequency	Start frequency of the X-Axis
FreqStep	Frequency	Frequency step of the X-Axis
SgramID	Enum	Optional: RMS, PPk, MPk, Avg, Smp
SgramElements	ULong	Number of SgramID Elements
TStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TStampFractional	ULong	Nanoseconds fractional part
FrameNumber	ULong	Frame Number
Overdriven	Bool	Overdriven flag
NotRealtime	Bool	NotRealtime flag
LevelValues	FlexArray[Float]	List of power levels in actual unit

### Examples

//Queries all valid frames (Spectrogram with 3 PPk frames)

SGRAM:DATA:ALL?

0,1551350976,656297236,3,INFO,5,3,3,51,90000000,400000,PPk,56,1551350976,656297236,1,0,0,-61.75,-62.97,-66.98,-70.65,-67.08,-62.94,-62.85,-67.11,-72.30,-72.86,-69.55,-65.74,-64.90,-68.04,-73.39,-74.19,-73.69,-67.28,-59.02,-55.28,-56.27,-62.74,-72.29,-76.86,-77.36,-77.34,-74.43,-68.39,-65.20,-65.32,-66.61,-67.33,-68.37,-70.48,-72.52,-67.13,-60.25,-57.80,-58.89,-62.72,-66.82,-70.81,-74.76,-75.00,-73.59,-73.44,-75.19,-76.49,-76.16,-76.04,-76.03,PPk,56,1551350975,656297236,2,0,0,-61.55,-63.05,-67.30,-70.40,-67.12,-62.79,-62.84,-67.18,-72.26,-73.02,-69.91,-65.75,-64.77,-67.48,-73.62,-75.53,-73.93,-67.51,-59.08,-55.36,-56.45,-62.65,-72.68,-76.24,-76.72,-77.14,-74.43,-68.51,-65.27,-65.38,-66.41,-67.56,-68.38,-70.40,-72.57,-67.18,-60.35,-57.72,-58.91,-62.76,-66.93,-71.28,-75.70,-75.10,-73.56,-73.79,-75.40,-76.21,-76.47,-76.37,-75.72,PPk,56,1551350974,656297236,3,0,0,-61.48,-62.89,-66.74,-70.08,-66.73,-62.54,-62.43,-66.89,-71.95,-72.91,-69.88,-65.49,-64.56,-67.44,-73.59,-75.40,-74.03,-67.42,-58.81,-54.80,-56.13,-62.89,-72.77,-76.40,-76.79,-77.19,-73.48,-68.31,-65.26,-65.35,-66.46,-67.34,-68.75,-70.34,-72.02,-67.32,-60.02,-57.00,-58.47,-62.52,-66.60,-71.03,-75.18,-75.33,-73.35,-73.55,-75.52,-76.50,-76.27,-76.05,-75.62

### SGRam:DATA:FRAME:COUNT?

Queries the number of valid spectrogram frames

#### Result Parameter

Parameter	Type	Description
Value	ULong	Number of valid frames

### Examples

//Queries the number of frames

SGR:DATA:FRAM:COUN?

450

### SGRam:DATA:UPDate?

Update is available in RUN and HOLD state. (see DATA:UPDate Commands)

#### Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong,Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

#### Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.

## Examples

//Spectrogram data update with old scancount = 10 and timeout = 100ms

SGR:DATA:UPD? 10,100ms

559900

//Forces Spectrogram data update because we do not have old scannumber

SGRam:DATA:UPDate?

558630

//Spectrogram data update with old scannumber = 0

SGRam:DATA:UPDate? 0

558980

//Forces Spectrogram data update with timeout = 100ms

SGRam:DATA:UPDate? NaN,100ms

559430

## SGRam:DETECTOR<Number>

Sets the spectrogram detector configuration

### Suffixes

Parameter	Type	Description
Number	UShort	Detector Number 1...3

### Command Parameter

Parameter	Type	Description
Value	Enum	Detector1 (Off,PPk), Detector2 (Off,RMS), Detector3 (Off, MPk, Smp, Avg)

## Examples

//Sets detector 3 to minus peak

SGR:DET3 MPk

## SGRam:DETECTOR<Number>?

Queries the spectrogram detector configuration

### Suffixes

Parameter	Type	Description
Number	UShort	Detector Number 1...3

### Result Parameter

Parameter	Type	Description
Value	Enum	Detector1 (Off, PPk, CPk), Detector2 (Off, RMS), Detector3 mode (Off, MPk, Smp, Avg)

### Examples

//Queries detector 3 configuration

SGR:DET3?

MPk

## SGRam:DETECTOR:VISible

Sets the spectrogram detector configuration

### Command Parameter

Parameter	Type	Description
Value	Enum	Detector (PPk, RMS, MPk, Smp, Avg)

### Examples

//Sets MPk as visible detector

SGR:DET:VIS MPk

//Sets PPk as visible detector

SGRam:DETECTOR:VISible PPk

## SGRam:DETECTOR:VISible?

Queries the spectrogram detector configuration

### Result Parameter

Parameter	Type	Description
Value	Enum	Detector (PPk, RMS, MPk, Smp, Avg)

### Examples

//Queries detector configuration

SGRam:DETECTOR:VISible?

MPk

## SGRam:FRAME:COUNT

Sets the maximum number of spectrogram frames

### Command Parameter

Parameter	Type	Description
Value	ULong	Max number of Spectrogram Frames

### Examples

//Sets number of spectrogram frames to 101

SGRam:FRAME:COUNT 101



## SGRam:FRAMe:COUNt?

Queries the maximum number of spectrogram frames

### Result Parameter

Parameter	Type	Description
Value	ULong	Max number of Spectrogram Frames

### Examples

//Queries number of spectrogram frames

SGR:FRAM:COUN?

101

## SGRam:SCAN:NUMBer?

Queries the spectrogram scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

### Examples

//Queries the current scan number

SGR:SCAN:NUMB?

1814

## Narda Command Reference - MAP

Map subsystem

### MAP:LOCalization:AREA

Sets the localization area.

#### Command Parameter

Parameter	Type	Description
TopLeftLat	Double	Top left latitude in degree (decimal)
TopLeftLon	Double	Top left longitude in degree (decimal)
BottomRightLat	Double	Bottom right latitude in degree (decimal)
BottomRightLon	Double	Bottom right longitude in degree (decimal)

#### Examples

//Sets the localization area.

MAP:LOCalization:AREA 48.548, 9.095, 48.368, 9.366

### MAP:LOCalization:AREA?

Queries the localization area.

#### Result Parameter

Parameter	Type	Description
TopLeftLat	Double	Top left latitude in degree (decimal)
TopLeftLon	Double	Top left longitude in degree (decimal)
BottomRightLat	Double	Bottom right latitude in degree (decimal)
BottomRightLon	Double	Bottom right longitude in degree (decimal)

#### Examples

//Queries the localization area.

MAP:LOCalization:AREA?

48.548, 9.095, 48.368, 9366

### MAP:LOCalization:BEARing:ERRor

Sets the bearing error for the localization

## Command Parameter

Parameter	Type	Description
Value	Float	Bearing error in degree 0°...20°

## Examples

//Sets the bearing error for the localization to 7.47 degree

MAP:LOC:BEAR:ERR 7.47

## MAP:LOCalization:BEARing:ERRor?

Queries the bearing error for the localization

## Result Parameter

Parameter	Type	Description
Value	Float	Bearing error in degree 0°...20°

## Examples

//Queries the bearing error for the localization

MAP:LOCalization:BEARing:ERRor?

16.2

## MAP:LOCalization:DATA?

Queries the localization data

## Result Parameter

Parameter	Type	Description
ValidFlag	Bool	Valid flag of the localization data
BearingCount	ULong	Number of bearings used for the localization
Lat	Float	Latitude of the localization result in degree (decimal)
Lon	Float	Longitude of the localization result in degree (decimal)
SemiMajorAxis	Float	Length of the semi-major axis of the error ellipse in m
SemiMinorAxis	Float	Length of the semi-minor axis of the error ellipse in m
Angle	Float	Angle of the semi-major axis with respect to north in degree
Frequency	Double	Frequency of the localization

## Examples

//Queries the localization data

MAP:LOCalization:DATA?

1,17,48.47213,9.217754,845.9222,593.3905,147.925,433000000

### MAP:LOCalization:FREQuency:TUNE:LIST?

Queries the available Ftunes for localization

#### Result Parameter

Parameter	Type	Description
Value	FlexArray[Double]	List frequencies for localization in Hz

#### Examples

//Queries the available Ftune for localization

MAP:LOC:FREQ:TUNE:LIST?

200000000.0,433000000.0,150000000.0

### MAP:LOCalization:FREQuency:TUNE:SElect

Sets the frequency of the localization

#### Command Parameter

Parameter	Type	Description
Value	Frequency	Frequency (Hz, kHz, MHz, GHz)

#### Examples

//Sets the frequency of the localization to 200 MHz

MAP:LOC:FREQ:TUNE:SEL 200 MHz

### MAP:LOCalization:FREQuency:TUNE:SElect?

Queries the frequency of the localization

#### Result Parameter

Parameter	Type	Description
Value	Double	Frequency of the localization in Hz

#### Examples

//Queries the selected Ftune of the localization

MAP:LOC:FREQ:TUNE:SEL?

200000000

### MAP:LOCalization:LOS:PROBability

Sets the line of sight probability for the localization

#### Command Parameter

Parameter	Type	Description
Value	Float	Line of sight probability in percent 0%,1%...100%

## Examples

//Sets the line of sight probability for the localization

MAP:LOC:LOS:PROB 50.0

## MAP:LOCalization:LOS:PROBability?

Queries the line of sight probability for the localization

### Result Parameter

Parameter	Type	Description
Value	Float	Line of sight probability in percent 0%,1%...100%

## Examples

//Queries the line of sight probability for the localization

MAP:LOCalization:LOS:PROBability?

60.0

## MAP:LOCalization:MIN:DFQuality

Sets the minimum DF quality for a bearing. The flag MAP:LOCalization:MIN:DFQuality:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:DFQuality to take effect!

### Command Parameter

Parameter	Type	Description
DFMin	Float	minimum DF quality 0%, 5%, ..., 100%

## Examples

//Sets the min DF quality to 80 percent.

MAP:LOCalization:MIN:DFQuality 80

## MAP:LOCalization:MIN:DFQuality?

Queries the minimum DF quality for a bearing to be taken into account by the localization. The flag MAP:LOCalization:MIN:DFQuality:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:DFQuality to take effect!

### Result Parameter

Parameter	Type	Description
DFMin	Float	minimum DF quality 0%, 5%, ..., 100%

## Examples

//Queries the min DF quality

MAP:LOCalization:MIN:DFQuality?

60

### MAP:LOCalization:MIN:DFQuality:ENABLE

Enables/Disables minimum DF quality criteria for bearings.

#### Command Parameter

Parameter	Type	Description
Value	Bool	min DF quality state

#### Examples

//Enables min DF quality criteria

MAP:LOC:MIN:DFQ:ENAB ON

### MAP:LOCalization:MIN:DFQuality:ENABLE?

Queries the minimum DF quality criteria state.

#### Result Parameter

Parameter	Type	Description
Value	Bool	min DF quality state

#### Examples

//Queries min DF quality criteria state

MAP:LOC:MIN:DFQ:ENAB?

1

### MAP:LOCalization:MIN:SPEed

Sets the min. Speed parameter for the localization. The flag MAP:LOCalization:MIN:SPEed:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:SPEed to take effect!

#### Command Parameter

Parameter	Type	Description
Value	Speed	min. Speed in m/s, km/h or mi/h

#### Examples

//Sets the min. Speed parameter for the localization to 20 km/h

MAP:LOC:MIN:SPE 20 km\_h

### MAP:LOCalization:MIN:SPEed?

Queries the min. Speed parameter for the localization. The flag MAP:LOCalization:MIN:SPEed:ENABLE must be enabled for the setting of MAP:LOCalization:MIN:SPEed to take effect!

#### Result Parameter

Parameter	Type	Description
Value	Float	min. Speed in m/s

## Examples

//Queries the min. Speed parameter for the localization in m/s

MAP:LOCalization:MIN:SPEed?

2.777

## MAP:LOCalization:MIN:SPEed:ENABLE

Enables/Disables minimum speed criteria for bearings to be taken into account by the localization.

### Command Parameter

Parameter	Type	Description
Value	Bool	min Speed state

## Examples

//Enables min Speed criteria

MAP:LOC:MIN:SPE:ENAB ON

## MAP:LOCalization:MIN:SPEed:ENABLE?

Queries the minimum speed criteria state for bearings to be taken into account by the localization.

### Result Parameter

Parameter	Type	Description
Value	Bool	min Speed state

## Examples

//Queries the min Speed criteria state

MAP:LOC:MIN:SPE:ENAB?

1

## MAP:LOCalization:MIN:SPEed:LIST?

Queries the list of possible values for the min. Speed parameter

### Result Parameter

Parameter	Type	Description
Value	FlexArray[Float]	List of min. Speed values in m/s

## Examples

//Queries the list of possible values for the min. Speed parameter in m/s

MAP:LOC:MIN:SPE:LIST?

0.277,0.555,0.833,1.388,2.777,5.555,8.333,13.888

//Queries the list of possible values for the min. Speed parameter in m/s

MAP:LOCalization:MIN:SPEed:LIST?

0.277,0.555,0.833,1.388,2.777,5.555,8.333,13.888

### MAP:LOCalization:RECORD:BEARing

Starts/Stops recording of bearings

#### Command Parameter

Parameter	Type	Description
Value	Bool	Type ON,OFF

#### Examples

//Starts recording

MAP:LOC:REC:BEAR ON

### MAP:LOCalization:RECORD:BEARing?

Queries the recording of bearings state

#### Result Parameter

Parameter	Type	Description
Value	Bool	Type ON,OFF

#### Examples

//Queries the recording state

MAP:LOCalization:RECORD:BEARing?

OFF

### MAP:LOCalization:RECORD:LIST?

Queries the available records for localization

#### Result Parameter

Parameter	Type	Description
Value	FlexArray[String]	List of records for localization

#### Examples

//Queries the available records for localization

MAP:LOC:REC:LIST?

RecordA, RecordB, RecordC

### MAP:LOCalization:RECORD:SElect

Selects the record used for localization

#### Command Parameter

Parameter	Type	Description
RecordName	String	Name of the record



## Examples

//Selects the record 'MyRecord' for localization

MAP:LOC:REC:SEL 'MyRecord'

## MAP:LOCalization:RECORD:SElect?

Queries the record used for localization

### Result Parameter

Parameter	Type	Description
RecordName	String	Name of the selected record

## Examples

//Queries the selected record for localization

MAP:LOCalization:RECORD:SElect?

'MyRecord'

## MAP:LOCalization:RESolution

Sets the resolution of the localization area

### Command Parameter

Parameter	Type	Description
Value	Enum	Resolution (VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW)

## Examples

//Sets the resolution of the localization area to HIGH

MAP:LOC:RES HIGH

## MAP:LOCalization:RESolution?

Queries the resolution of the localization area

### Result Parameter

Parameter	Type	Description
Value	Enum	Resolution (VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW)

## Examples

//Queries the resolution of the localization area

MAP:LOCalization:RESolution?

NORMAL

### MAP:LOCalization:START

Starts the localization

#### Examples

//Starts the localization

MAP:LOCalization:START

//Starts the localization

MAP:LOC:STAR

### MAP:LOCalization:STATe?

Queries the localization state

#### Result Parameter

Parameter	Type	Description
Value	Enum	START, STOP

#### Examples

//Queries the localization state

MAP:LOC:STAT?

STOP

//Queries the localization state

MAP:LOCalization:STATe?

START

### MAP:LOCalization:STOP

Stops the localization

#### Examples

//Stops the localization

MAP:LOCalization:STOP

//Stops the localization

MAP:LOC:STOP

### MAP:LOCalization:USE:CASE:PRESet

Sets the use case preset for the localization

#### Command Parameter

Parameter	Type	Description
Value	Enum	Use case preset (CITY_VEHICLE_GNSS, CITY_FIXED_SITE, FREE_FIELD_VEHICLE_GNSS , FREE_FIELD_FIXED_SITE)

## Examples

//Sets the use case preset for localization to CITY\_VEHICLE\_GNSS

MAP:LOC:USE:CASE:PRESet CITY\_VEHICLE\_GNSS

## MAP:LOCalization:USE:CASE:PRESet?

Queries the use case preset for the localization

### Result Parameter

Parameter	Type	Description
Value	Enum	Use case preset (CITY_VEHICLE_GNSS, CITY_FIXED_SITE, FREE_FIELD_VEHICLE_GNSS, FREE_FIELD_FIXED_SITE)

## Examples

//Queries the use case preset for localization

MAP:LOCalization:USE:CASE:PRESet?

FREE\_FIELD\_VEHICLE\_GNSS

## MAP:MODE

Sets the map mode

### Command Parameter

Parameter	Type	Description
Mode	Enum	Mode (NONE, DISCRETE_LOCALIZATION, CONTINUES_LOCALIZATION, MEASUREMENT_SITES, COVERAGE, LOCALIZATION_RESULTS)

## Examples

//Sets the map mode to CONTINUES\_LOCALIZATION

MAP:MODE CONTINUES\_LOCALIZATION

## MAP:MODE?

Queries the map mode

### Result Parameter

Parameter	Type	Description
Mode	Enum	Mode (NONE, DISCRETE_LOCALIZATION, CONTINUOUS_LOCALIZATION, MEASUREMENT_SITES, COVERAGE, LOCALIZATION_RESULTS)

### Examples

//Queries the current map mode

MAP:MODE?

NONE

## Narda Command Reference - HORizontal

HORizontal Scan Subsystem (undefined Commands)

### HORizontal:SCAN:ADD:VALue

Horizontal scan add discrete value

#### Examples

```
//Adds current value horizontal scan
```

```
HORizontal:SCAN:ADD:VALue
```

### HORizontal:SCAN:CALC:RESult

Horizontal scan calculate bearing

#### Examples

```
//Horizontal scan calculate result
```

```
HORizontal:SCAN:CALC:RESult
```

### HORizontal:SCAN:DATA?

Queries the Horizontal scan results

#### Result Parameter

Parameter	Type	Description
ValidFlag	Bool	Valid flag of the localization data.
Overdriven	Bool	Overdriven flag
LostFrames	Bool	LostFrames flag
Uncertain	Bool	Uncertain flag
DetectorValue	Float	Detector value
Azimuth	Float	Azimuth in degree.
Elevation	Float	Elevation in degree.
Roll	Float	Roll in degree.

#### Examples

```
//Queries the horizontal scan results
```

```
HORizontal:SCAN:DATA?
```

```
1,0,0,0,-85.63,218.799,-12.572,-1.902
```

### HORizontal:SCAN:DELeTe:VALue

Horizontal scan delete last value

#### Examples

```
//Deletes last horizontal scan value
```

```
HORizontal:SCAN:DELeTe:VALue
```

### HORizontal:SCAN:DETEctor

Sets the Horizontal scan detector

#### Command Parameter

Parameter	Type	Description
Value	Enum	Horizontal scan detector (RMS, CRMS, PPK, CPK)

#### Examples

//Sets horizontal scan detector to RMS

HORizontal:SCAN:DETEctor RMS

### HORizontal:SCAN:DETEctor?

Queries the Horizontal scan detector

#### Result Parameter

Parameter	Type	Description
Value	Enum	Horizontal scan detector (RMS, CRMS, PPK, CPK)

#### Examples

//Queries the horizontal scan detector

HORizontal:SCAN:DETEctor?

RMS

### HORizontal:SCAN:MANual:CORRection

Horizontal scan manual correction

#### Examples

//Horizontal scan manual correction

HORizontal:SCAN:MANual:CORRection

### HORizontal:SCAN:RESet

Horizontal scan reset

#### Examples

//Resets horizontal scan values

HORizontal:SCAN:RESet

### HORizontal:SCAN:RESet:MAX

Horizontal scan reset max

## Examples

//Resets horizontal scan max values

HORizontal:SCAN:RESet:MAX

## HORizontal:SCAN:START

Horizontal scan start

## Examples

//Starts horizontal scan

HORizontal:SCAN:STARt

## HORizontal:SCAN:STATe?

Queries the Horizontal scan state

## Result Parameter

Parameter	Type	Description
Value	Enum	Horizontal scan state ( INIT, START, RESET_MAX, ADD_DISCRETE, ADD_DISCRETE_MAX, DELETE_LAST_VALUE, CALCULATE_BEARING, MANUAL_BEARING )

## Examples

//Queries horizontal scan state

HORizontal:SCAN:STATe?

START

## HORizontal:SCAN:STOP

Horizontal scan stop

## Examples

//Stops horizontal scan

HORizontal:SCAN:STOP

## HORizontal:SCAN:TYPE

Sets the Horizontal scan type

## Command Parameter

Parameter	Type	Description
Value	Enum	Horizontal scan type (CONTINUOUS, DISCRETE, DISCRETE_MAX_HOLD)

### Examples

//Sets horizontal scan type to CONTINUOUS

HORizontal:SCAN:TYPE CONTINUOUS

### HORizontal:SCAN:TYPE?

Queries the Horizontal scan type

### Result Parameter

Parameter	Type	Description
Value	Enum	Horizontal scan type (CONTINUOUS, DISCRETE, DISCRETE_MAX_HOLD)

### Examples

//Queries the horizontal scan type

HORizontal:SCAN:TYPE?

CONTINUOUS



## Narda Command Reference - PERSistence

Persistence subsystem

### PERSistence:DATA:ALL?

Queries a persistence image. Caution: You must skip unused and unknown IDs and data elements while parsing to be compatible with future extensions. (see DATA:ALL? commands)

#### Query Parameter

Parameter	Type	Description
Optional	Optional[Enum]	Reserved for future use

#### Result Parameter

Parameter	Type	Description
TimeStampSyncFlag	Bool	Timestamp Sync Flag
TimeStampSeconds	ULong	Seconds since 1.1.1970 0:00 Greenwich Mean Time
TimeStampFractional	ULong	Nanoseconds fractional part
ScanNumber	ULongLong	Scan Number
ColumnCount	ULong	Number of frequency bins along the x-axis, i.e. number of columns in the probability matrix
RowCount	ULong	Number of power level bins along the y-axis, i.e. number of rows in the probability matrix
FreqStart	Frequency	Start frequency of the X-Axis
FreqStep	Frequency	Frequency step of the X-Axis
PersistenceType	Enum	Optional: Max, Avg
Overdriven	Bool	Overdriven flag
NotRealtime	Bool	NotRealtime flag
PersistenceValues	FlexArray[Float]	List of probability values in range from 0 (0 % probability) to 1 (100% probability) (see "SCPI Example - Persistence" in the Appendix for more detailed information)

#### Examples

//Queries the current persistence data

PERS:DATA:ALL?

0,1592475133,816676044,4139,801,601,73338084,50000,MAX,0,0,0.002,0.0025,0.0023333,0.002,0.0020833,0.0019167, ...

### PERSistence:DATA:COLUMN:COUNT?

Queries the number of columns of a persistence image

### Result Parameter

Parameter	Type	Description
Value	ULong	Number of persistence columns

### Examples

//Queries the number of persistence columns

PERS:DATA:COL:COUN?

801

## PERSistence:DATA:FREQuency:STARt?

Queries persistence data start frequency

### Result Parameter

Parameter	Type	Description
Value	Frequency	Start frequency of the X-Axis

### Examples

//Queries the start frequency for the persistence results

PERS:DATA:FREQ:STAR?

87400000

## PERSistence:DATA:FREQuency:STEP?

Queries persistence data frequency step

### Result Parameter

Parameter	Type	Description
Value	Frequency	Frequency step of the X-Axis

### Examples

//Queries the frequency step for the persistence results

PERS:DATA:FREQ:STEP?

200000

## PERSistence:DATA:OVERdriven?

Queries the persistence data overdriven flag

### Result Parameter

Parameter	Type	Description
Value	Bool	Overdriven flag

## Examples

//Queries the overdriven flag

PERS:DATA:OVER?

0

## PERSistence:DATA:REALtime?

Queries the pesistence data realtime flag

### Result Parameter

Parameter	Type	Description
Value	Bool	Realtime flag

## Examples

//Queries the realtime flag

PERS:DATA:REAL?

1

## PERSistence:DATA:ROW:COUNT?

Queries the number of rows of a persistence image

### Result Parameter

Parameter	Type	Description
Value	ULong	Number of persistence rows

## Examples

//Queries the number of persistence rows

PERS:DATA:ROW:COUN?

601

## PERSistence:DATA:UPDate?

Update is available in RUN and HOLD mode (see DATA:UPDate Commands)

### Query Parameter

Parameter	Type	Description
Optional	Optional[ULongLong, Timespan]	ScanNumber of last queried measurement data or NaN to force an update, Timeout value for update data (s, ms)

### Result Parameter

Parameter	Type	Description
ScanNumber	ULongLong	ScanNumber of current measurement data.

### Examples

//Persistence data update with old scantcount = 10 and timeout = 100ms

PERS:DATA:UPD? 10,100ms

559900

//Forces Persistence data update because we do not have old scantnumber

PERS:DATA:UPDate?

558630

//Persistence data update with old scantnumber = 0

PERS:DATA:UPDate? 0

558980

//Forces Persistence data update with timeout = 100ms

PERS:DATA:UPDate? NaN,100ms

559430

## PERSistence:MEASurement:TIME

Measurement time for persistence

### Command Parameter

Parameter	Type	Description
Value	Timespan	Measurement time (h, m, s, ms)

### Examples

//Sets the persistence measurement time to 1s

PERS:MEAS:TIME 1 s

## PERSistence:MEASurement:TIME?

Queries the persistence measurement time

### Result Parameter

Parameter	Type	Description
Value	Timespan	Measurement time in s

### Examples

//Queries the persistence measurement time

PERS:MEAS:TIME?

0.01

## PERSistence:SCAN:COUNT

Sets the persistence scan count

### Command Parameter

Parameter	Type	Description
Value	ULong	Scan count

### Examples

//Sets the scan count to 50

PERS:SCAN:COUN 50

### PERSistence:SCAN:COUNT?

Queries the current scan count

### Result Parameter

Parameter	Type	Description
Value	ULong	Scan count

### Examples

//Queries the current scan count

PERS:SCAN:COUN?

50

### PERSistence:SCAN:NUMBER?

Queries the persistence scan number

### Result Parameter

Parameter	Type	Description
Value	ULongLong	Scan number

### Examples

//Queries the current scan number

PERS:SCAN:NUMB?

1814

### PERSistence:TYPE

Sets the persistence type

### Command Parameter

Parameter	Type	Description
Value	Enum	Max, Avg

### Examples

//Sets the persistence type to Max

PERS:TYPE Max

PERSistence:TYPE?

Queries the persistence type

Result Parameter

Parameter	Type	Description
Value	Enum	Max, Avg

Examples

```
//Queries the persistence type
PERS:TYPE?
Avg
```

## Narda Command Reference - Deprecated

Deprecated subsystem

### BEARing:PERSistence

Deprecated Command: Use BEARing:POST:AVG instead

#### Command Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time (h, m, s, ms)

#### Examples

//Deprecated Command: Sets the post averaging time to 10ms

BEAR:PERS 0.01

### BEARing:PERSistence?

Deprecated Command: Use BEARing:POST:AVG? instead

#### Result Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time in s

#### Examples

//Deprecated Command: Queries the post averaging time

BEAR:PERS?

0

### LEVelmeter:DETEctor:PERSistence

Deprecated Command: Use LEVelmeter:POST:AVG instead

#### Command Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time (h, m, s, ms)

#### Examples

//Deprecated Command: Sets the post averaging time to 10ms

LEV:DET:PERS 10 ms

### LEVelmeter:DETEctor:PERSistence?

Deprecated Command: Use LEVelmeter:POST:AVG? instead

### Result Parameter

Parameter	Type	Description
Value	Timespan	Post averaging time in s

### Examples

//Deprecated Command: Queries the post averaging time

LEV:DET:PERS?

0.01

### LEVelmeter:TVIDeo

Deprecated Command: Use LEVelmeter:PRE:AVG instead

### Command Parameter

Parameter	Type	Description
Value	Timespan	Pre averaging time (h, m, s, ms), 0 = OFF

### Examples

//Deprecated Command: Sets the pre averaging time to 10ms

LEV:TVID 10 ms

### LEVelmeter:TVIDeo?

Deprecated Command: Use LEVelmeter:PRE:AVG? instead

### Result Parameter

Parameter	Type	Description
Value	Timespan	Pre averaging time in s, 0 = OFF

### Examples

//Deprecated Command: Queries the pre averaging time

LEV:TVID?

0.01

### [SENSe:]APPLication:QUIT

Deprecated Command: Use SYSTem:SHUTdow instead

### Examples

//Deprecated Command: This command shuts down the instrument

APPL:QUIT

### [SENSe:]ATTenuator:ENTRy:MODE

Deprecated Command: Has no effect anymore



## Command Parameter

Parameter	Type	Description
Value	Enum	Entry mode (ATTENUATOR,REFERENCE_LEVEL)

## Examples

//Deprecated Command: Sets entry mode to REFERENCE\_LEVEL

ATT:ENTR:MODE REFERENCE\_LEVEL

## [SENSe:]ATTenuator:ENTRy:MODE?

Deprecated Command: Has no effect anymore

## Result Parameter

Parameter	Type	Description
Value	Enum	Entry mode (ATTENUATOR,REFERENCE_LEVEL)

## Examples

//Deprecated Command: Queries the entry mode

ATT:ENTR:MODE?

REFERENCE\_LEVEL

## [SENSe:]REFerence:LEVel:ENTRy:MODE

Deprecated Command: Has no effect anymore

## Command Parameter

Parameter	Type	Description
Value	Enum	Entry mode (ATTENUATOR,REFERENCE_LEVEL)

## Examples

//Deprecated Command: Sets entry mode to REFERENCE\_LEVEL

REF:LEV:ENTR:MODE REFERENCE\_LEVEL

## [SENSe:]REFerence:LEVel:ENTRy:MODE?

Deprecated Command: Has no effect anymore

## Result Parameter

Parameter	Type	Description
Value	Enum	Entry mode (ATTENUATOR,REFERENCE_LEVEL)

### Examples

//Deprecated Command: Queries the entry mode

REF:LEV:ENTR:MODE?

REFERENCE\_LEVEL

### [SENSe:]RUN:SINGLE:OVERlap

Deprecated Command: Use RUN:SINGLE instead

#### Command Parameter

Parameter	Type	Description
Optional	Optional[Timespan]	Timeout value for initializing and single measurement (s, ms)

### Examples

//Deprecated Command: Starts a single measurement as non-blocking command with a timeout of 10 seconds for initializing and measurement

RUN:SING:OVER 10s

//Deprecated Command: Starts a single measurement as a non-blocking command with default timeout

SENSe:RUN:SINGLE:OVERlap

//Deprecated Command: Starts a single measurement as a blocking command until initializing and measurement has finished

RUN:SING:OVER;\*WAI

### [SENSe:]STOP:MODE

Deprecated Command: Use STOP and HOLD instead

#### Command Parameter

Parameter	Type	Description
Value	Enum	Stop mode (STOP,HOLD)

### Examples

//Deprecated Command: Select the hold mode

STOP:MODE HOLD

//Deprecated Command: Select the stop mode

SENSe:STOP:MODE STOP

### [SENSe:]STOP:MODE?

Deprecated Command: Use STOP and HOLD instead

#### Result Parameter

Parameter	Type	Description
Value	Enum	Stop mode (STOP,HOLD)

### Examples

//Deprecated Command: Queries the stop mode

STOP:MODE?

HOLD



# 11 Appendix: Examples

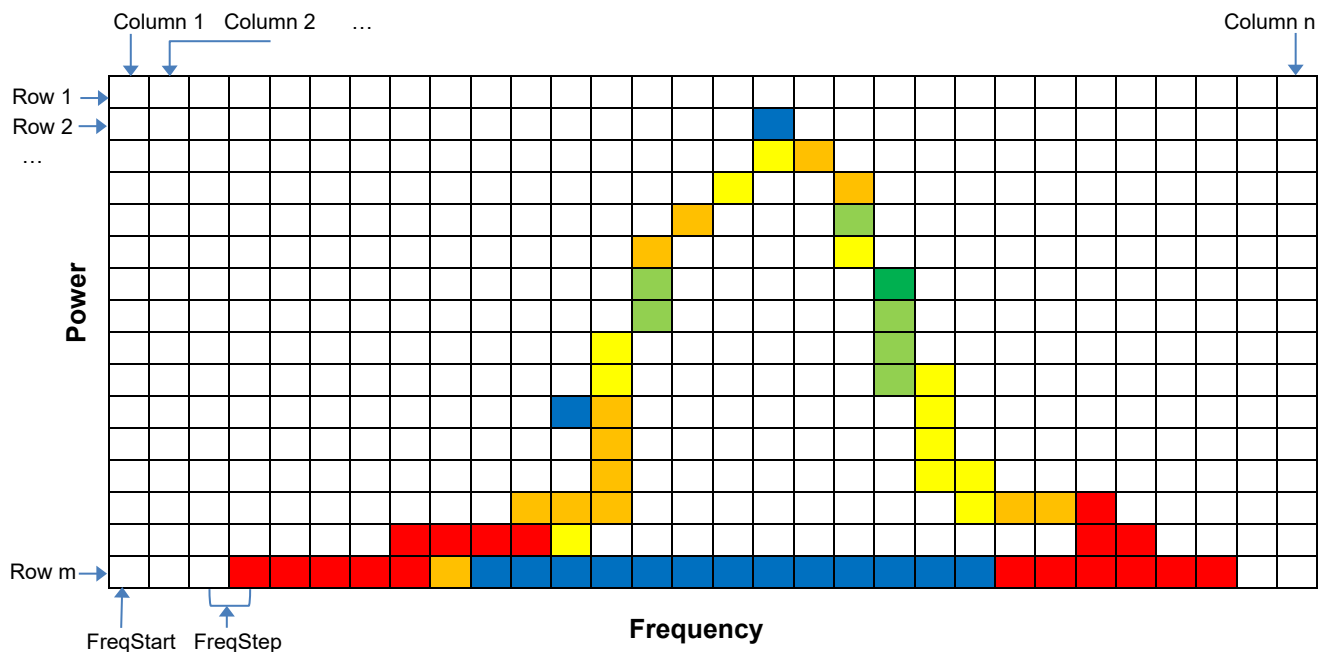
## SCPI Example – RT Spectrum

//Setup Task and Views		
← *RST		//Optional Reset device.
		//This is not needed when using TASK:NEW?
	SYSTem:ERRor:CODE:ALL?	//Check for errors
→ 0		//all OK
← TASK:NEW? 'RT SPECTRUM'		//Add RT Spectrum task, delete other Tasks
→ "RT Spectrum"		
← TASK:SElect 'RT Spectrum'		//This is only necessary, if RT Spectrum task still exists.
//Setup measurement parameters		
← SENSE:ATTenuator 10 dB		//Set attenuator to 10 dB
	SPECTrum:FREQuency:STOP 1.04e9	//Set stop frequency to 1.04 GHz
← SPECTrum:FREQuency:STARt 1e9		//Set start frequency to 1 GHz
← SPECTrum:RBW 100e3		//Set RBW to 100 kHz
← SPECTrum:MEAS:TIME 10ms		//Set the measurement time to 10 ms
← SPECTrum:TRACe:ENABLE AvR,ON		//Enable trace AvR
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//all OK
//Measurement sequence		
← RUN:CONT;HOLD		//Hold data in readout memory for synchronized readout
//Acquire first spectrum data to get trace basic information		
← SPECTrum:DATA:UPDate?		//Wait until new spectrum data is aquired
→ 3		//Returns current ScanNumber (e.g. 3)
← SPECTrum:DATA:COUNT?		//Request number of bins
→ 801		
← SPECTrum:DATA:FREQuency:STARt?		//Request SPECTrum data start frequency
→ 1000000000		
← SPECTrum:DATA:FREQuency:STEP?		//Request SPECTrum data frequency step
→ 50000		
//Measurement loop for fast acquisition of spectrum data.		
← SPEC:DATA:UPD? 3,500ms		//Wait until Scan Number is greater then last Scan Number
		//or timeout occurs
→ 4		//Returns current ScanNumber (e.g. 4)
← SPECTrum:DATA:OVERdriven?		//Check for overdriven
→ 0		//not overdriven
← SPECTrum:DATA:REALtime?		//Check for real-time
→ 1		//it is real-time
← SPEC:DATA:LEVel? AvR		//Request AvR trace data
→ -89.54,-93.91,-99.58,-100.34,-100.44,-100.58,...-98.47		//the level values depends on the selected unit

## Use of SPECTrum:DATA result

Index	Frequency	AvR Value
1	1000000000 (start frequency)	-89.54
2	1000000000 + 50000 (+ frequency step)	-93.91
3	1000000000 + 50000 + 50000	-99.58
4	1000000000 + 50000 + 50000 + 50000	-100.34
...	...	...
161	1000000000 + 50000 + ... + 50000	-98.47

## SCPI Example – Persistence



Probability of occurrence:

red = Highest probability of occurrence, blue = Lowest probability of occurrence

From the command `PERSistence:DATA:ALL?`, the last parameter, "PersistenceValues", in the respond returns a list of all probability values in a range from 0 (0% probability of occurrence) to 1 (100% probability of occurrence).

The "ColumnCount" parameter returns the number of columns in the probability matrix and the "RowCount" parameter returns the number of rows in the probability matrix.

With this information it is possible to build up the probability matrix in the following way:

"PersistenceValues" list:

- ☐ 1<sup>st</sup> probability value in the list correspond with Row 1 and Column 1 in the matrix: ( $a_{11}$ )
- ☐ 2<sup>d</sup> probability value in the list correspond with Row 1 and Column 2 in the matrix: ( $a_{12}$ )
- ☐ 3<sup>d</sup> probability value in the list correspond with Row 1 and Column 3 in the matrix: ( $a_{13}$ )

...

- ☐  $n^{\text{th}}$  probability value in the list correspond with Row 1 and Column n in the matrix: ( $a_{1n}$ )
- ☐ ( $n+1$ )<sup>th</sup> probability value in the list correspond with Row 2 and Column 1 in the matrix: ( $a_{21}$ )
- ☐ ( $n+2$ )<sup>th</sup> probability value in the list correspond with Row 2 and Column 2 in the matrix: ( $a_{22}$ )
- etc...

Spectrum parameters can be set with the SPECTrum:XXX-commands (see “Narda Command Reference – SPECTrum”)

//Setup Task and Views		
← *RST		//Optional Reset device.
		//This is not needed when using TASK:NEW?
SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//all OK
← TASK:NEW? 'RT SPECTRUM'		//Add RT Spectrum task, delete other Tasks
→ "RT Spectrum"		
← TASK:SELEct 'RT Spectrum'		//This is only necessary, if more tasks exists.
← VIEW:ADD? 1,BELOW,'PERSISTENCE'		//Add a Persistence view below the default spectrum view
← VIEW:LIST?		//Request a list of existing views in the current task
→ "PERSISTENCE",2,"SPECTRUM",1		
← VIEW:SEL 2		//Select the Persistence view with index 2

//Setup measurement and display parameters		
← SENSE:ATTenuator 10 dB		//Set attenuator to 10 dB
← SPECTrum:FREQuency:STOP 100 MHz		//Set stop frequency to 100 MHz
← SPECTrum:FREQuency:STARt 96 MHz		//Set start frequency to 96 MHz
← SPECTrum:RBW 10e3		//Set RBW to 10 kHz
← DISP:PERS:LRAN 100		//Set the maximum displayed level range to 100 dB
← DISP:PERS:LMAX -70		//Set the maximum displayed level value to -70 dBm
PERS:TYPE Max		//Set the Persistence type to Max
← PERS:MEAS:TIME 50ms		//Set the persistence measurement time to 50 ms
← PERS:SCAN:NUMB 1		//Set the scan number to 1
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//all OK

//Measurement sequence		
← RUN:CONT;HOLD		
← PERS:DATA:UPD? 0,500ms		//Wait until Scan Number is > 0 or timeout occurs
→ 4		//Returns current ScanNumber (e.g. 4)
← PERS:DATA:ALL?		//Request all data, last position gives the probability value-list
→ 0,1592475133,816676044,4139,801,601,73338084,50000,MAX,0,0,0.002,0.0025,0.0023333,...		
← PERS:DATA:UPD? 4,500ms		//Wait until Scan Number is greater then last Scan Number or timeout occurs
→ 5		//Returns current ScanNumber (e.g. 5)
← PERS:DATA:ALL?		//Request all data, last position gives the probability value-list

## SCPI Example – Automatic DF

//Setup Task and Views		
←	*RST	//Optional Reset device. //This is not needed when using TASK:NEW?
	SYST:ERR:CODE:ALL?	//Check for errors
→	0	//all OK
←	TASK:NEW? 'AUTO_DF'	//Add Auto DF task, delete other Tasks and request new task name
→	"Auto DF"	
←	TASK:SEL 'Auto DF'	//This is only necessary, if AutoDF task still exists.

//Setup Auto DF spectrum parameters		
←	SENSe:ATT 0 dB	//Set attenuator to 0 dB
←	SPEC:FREQ:CENT 446 MHz	//Set frequency (fcent = ftune)
←	SPEC:RBW 100e3	//Set RBW to 100 kHz
//Setup Auto DF bearing parameters		
←	BEAR:FREQ:TUNE 446 MHz	//Set tuning frequency (fcent = ftune)
←	BEAR:CBW 12.5e3	//Set channel bandwidth to 12.5 kHz
←	BEAR:MEAS:TIME 1ms	//Set bearing measurement time to 1 ms
←	SETT:MAGN:DECL 2.68	//Set magnetic declination (only required if compass is used)
←	BEAR:NORT:REF REFERENCE_MARK_DIRECTION	//Set north reference to antenna reference mark direction
←	BEAR:REF:MARK:DIR 0	//Set the deviation of the antenna reference mark direction to north in degree
←	BEAR:AZIM:CORR 0	//Set the azimuth correction in degree to 0°
←	BEAR:ing:DFSQuelch -80 dBm	//Set DF squelch to avoid bearing results based on noise.
←	BEAR:DFSQ:ENAB ON	//Switch DF squelch filter ON
←	BEAR:MIN:DFQ 80	//Set bearing minimum quality filter to 80%
←	BEAR:MIN:DFQ:ENAB ON	//Switch on DF minimum quality filter
←	BEAR:MIN:STAB:ENAB OFF	//Switch off min. stability filter
←	BEAR:DATA:VAL:ENAB 1	//Set BEAR:DATA:VAL:ENAB flag, so that DFSQuelch and MIN:DFQ becomes applied.
←	SYST:ERR:CODE:ALL?	//Check for errors
→	0	//all OK

//Measurement loop		
←	BEAR:FREQ:TUNE 446 MHz	//Set new tuning frequency if needed //(Spectrum->fcent equals Bearing->ftune)
←	RUN:CONT;HOLD	//Hold data in readout memory for synchronized readout
←	BEAR:DATA:UPD? 0,500ms	//Wait until Scan Number is greater then last Scan Number //or timeout occurs
→	3	//Returns current ScanNumber (e.g. 3)
←	BEAR:DATA:AZIM?	//Request azimuth value
→	272.9964	
←	BEAR:DATA:ELEV?	//Request elevation value
→	37.5	//
←	BEAR:DATA:DFQ?	//Request DF quality of bearing
→	82.45299	//e.g. 82.45299 %
←	BEAR:DATA:UPD? 3,500ms	//Wait until Scan Number is greater then last Scan Number //or timeout occurs
→	4	//Returns current ScanNumber (e.g. 4)
←	BEAR:DATA:AZIM?	//Request azimuth value
→	271.1	
←	BEAR:DATA:ELEV?	//Request elevation value
→	38	//
←	BEAR:DATA:DFQ?	//Request DF quality of bearing
→	81.237	//



## SCPI Example – VITA49 IQ Streaming



*From a IQStream:CBW greater than 5 MHz, it is recommended to use jumbo packages with 9014 bytes for all network components, otherwise data packets may be lost. In addition, a suitable Ethernet cable (e.g. CAT 7) must be used. The weakest component in the network determines the maximum possible performance. Network components are e.g. the network cards of SignalShark and the PC as well as routers or switches.*

//Setup Task and Views		
← *RST		//Optional Reset device.
		//This is not needed when using TASK:NEW?
	SYSTem:ERRor:CODE:ALL?	//Check for errors
→ 0		//All OK
← TASK:NEW? 'RT_STREAMING'		//Add RT Streaming task, delete other Tasks
→ "RT Streaming"		
← STReam:ADD? 'VITA49 IQ'		//Add a new streaming view
→ 1		
	STReam:Select 1	Select the added stream (view)
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Setup connection parameters		
← STReam:CONNECTION:IDN 12345		//Set stream identifier number
← STReam:CONNECTION:IDN?		//Check stream identifier number
→ 12345		
	STReam:CONNECTION:TYPE UDP SINGLECAST	//Set stream type to UDP SINGLECAST
	STReam:CONNECTION:ADDRESS "192.168.128.1"	//Set UDP streaming client address (e.g. a PC)
← STReam:CONNECTION:PORT 4444		//Set UDP streaming port
← STReam:CONNECTION:PORT?		//Check UDP streaming port
→ 4444		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Setup IQ-Streaming parameters		
← IQStream:CBW:LIST?		
→ 20000000,19200000,16000000,...		
← IQStream:CBW 750kHz		
← IQStream:CBW?		
→ 750000		
← IQStream:FREQUENCY:TUNE 97.7MHz		
← IQStream:OVERSAMPLING OFF		
← IQStream:PAYLOAD:LIMIT:LIST?		
→ 32,64,128,256,512,1024,2048		
← IQStream:PAYLOAD:LIMIT 2048		
← IQStream:PAYLOAD:FORMAT I16Q16		
← SYSTem:ERRor:CODE:ALL?		
→ 0		

//Open/close connection and start/stop streaming		
← STReam:CONNECTION:OPEN		//Open connection and start streaming
← STReam:CONNECTION:STATE?		//Check connection state
→ CONNECTED		
← STReam:CONNECTION:CLOSE		//Close connection and stop streaming
← STReam:CONNECTION:STATE?		//Check connection state
→ CLOSED		

## SCPI Example – VITA49 Spectrum Streaming via UDP Single Cast

Spectrum parameters can be set with the SPECTrum:XXX-commands (see “Narda Command Reference – SPECTrum”)

//Setup Task and Views		
←	*RST	//Optional Reset device. //This is not needed when using TASK:NEW?
	SYSTem:ERRor:CODE:ALL?	//Check for errors
→	0	//All OK
←	TASK:NEW? 'RT STREAMING'	//Add RT Streaming task, delete other Tasks
→	"RT Streaming"	
←	STReam:ADD? 'VITA49 SPECTRUM RMS'	//Add a new streaming view, here with RMS data
→	1	
	STReam:Select 1	Select the added stream (view)
←	SYSTem:ERRor:CODE:ALL?	//Check for errors
→	0	//All OK

//Setup connection parameters		
←	STReam:CONNection:IDN 12345	//Set stream identifier number
←	STReam:CONNection:IDN?	//Check stream identifier number
→	12345	
	STReam:CONNection:TYPE UDP_SINGLECAST	//Set stream type to UDP_SINGLECAST
	STReam:CONNection:ADDReSS "192.168.128.1"	//Set UDP streaming client address (e.g. a PC)
←	STReam:CONNection:PORT 4444	//Set UDP streaming port
←	STReam:CONNection:PORT?	//Check UDP streaming port
→	4444	
←	SYSTem:ERRor:CODE:ALL?	//Check for errors
→	0	//All OK

//Open/close connection and start/stop streaming		
←	STReam:CONNection:OPEN	//Open connection and start streaming
←	STReam:CONNection:STAt?	//Check connection state
→	CONNECTED	
←	STReam:CONNection:CLOSE	//Close connection and stop streaming
←	STReam:CONNection:STAt?	//Check connection state
→	CLOSED	

## SCPI Example – Audio Streaming via TCP

Demodulation parameters can be set with the DEMod:XXX-commands (see “Narda Command Reference – DEMod”)

//Setup Task and Views		
← *RST		//Optional Reset device.
		//This is not needed when using TASK:NEW?
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK
← TASK:SElect?		//Check which task is currently selected
→ "RT SPECTRUM", "RT Spectrum"		
← DEM:TYPE FM		//Set the demodulation type to FM
← DEM:TYPE?		//Check the set demodulation type
→ FM		
← DEM:TYPE:ENAB 1		//Switch on the set demodulation type
← DEM:TYPE:ENAB?		//Check demodulation type enable state
→ 1		
← DEM:CBW:LIST?		//Check available CBWs
→ 1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000, ... , 128,125,120,100,50,25		
← DEM:CBW 200 kHz		//Set the CBW to 200 kHz
← DEM:CBW?		//Check the CBW
→ 200000		
← DEM:FREQ:TUNE 97.7 MHz		//Set the demodulation tune frequency to 97.7 MHz
← DEM:FREQ:TUNE?		//Check the demodulation tune frequency
→ 97700000		
← STR:LIST?		//Check for available streams
→ "AUDIO_DEMOD",1		
← STR:SEL 1		//Select the audio demod stream (index 1)
← STR:SEL?		//Check selected stream
→ 1		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Setup connection parameters		
← STReam:CONNECTION:IDN 12345		//Set stream identifier number
← STReam:CONNECTION:IDN?		//Check stream identifier number
→ 12345		
← STR:CONN:TYPE TCP SERVER		//Set stream type to TCP SERVER
← STR:CONN:TYPE?		//Check stream type
→ TCP SERVER		
← STReam:CONNECTION:PORT 4444		//Set TCP streaming port
← STReam:CONNECTION:PORT?		//Check TCP streaming port
→ 4444		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Open/close connection and start/stop streaming		
← STReam:CONNECTION:OPEN		//Open connection and start streaming
← STReam:CONNECTION:STATe?		//Check connection state
→ CONNECTED		
← STReam:CONNECTION:CLOSE		//Close connection and stop streaming
← STReam:CONNECTION:STATe?		//Check connection state
→ CLOSED		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

## SCPI Example – Audio Streaming via UDP Single Cast

Demodulation parameters can be set with the DEMod:XXX-commands (see “Narda Command Reference – DEMod”)

//Setup Task and Views		
← *RST		//Optional Reset device. //This is not needed when using TASK:NEW?
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK
← TASK:SElect?		//Check which task is currently selected
→ "RT SPECTRUM", "RT Spectrum"		
← DEM:TYPE FM		//Set the demodulation type to FM
← DEM:TYPE?		//Check the set demodulation type
→ FM		
← DEM:TYPE:ENAB 1		//Switch on the set demodulation type
← DEM:TYPE:ENAB?		//Check demodulation type enable state
→ 1		
← DEM:CBW:LIST?		//Check available CBWs
→ 1000000,960000,800000,750000,640000,625000,600000,512000,500000,480000,400000,384000,320000, ... , 128,125,120,100,50,25		
← DEM:CBW 200 kHz		//Set the CBW to 200 kHz
← DEM:CBW?		//Check the CBW
→ 200000		
← DEM:FREQ:TUNE 97.7 MHz		//Set the demodulation tune frequency to 97.7 MHz
← DEM:FREQ:TUNE?		//Check the demodulation tune frequency
→ 97700000		
← STR:LIST?		//Check for available streams
→ "AUDIO_DEMOD",1		
← STR:SEL 1		//Select the audio demod stream (index 1)
← STR:SEL?		//Check selected stream
→ 1		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Setup connection parameters		
← STReam:CONNECTION:IDN 12345		//Set stream identifier number
← STReam:CONNECTION:IDN?		//Check stream identifier number
→ 12345		
← STReam:CONNECTION:TYPE UDP SINGLECAST		//Set stream type to UDP SINGLECAST
← STR:CONN:TYPE?		//Check stream type
→ UDP SINGLECAST		
← STReam:CONNECTION:ADDRESS "192.168.178.9"		//Set UDP streaming client address (e.g. a PC)
← STR:CONN:ADDR?		//Check streaming client address
→ "192.168.178.9"		
← STReam:CONNECTION:PORT 4444		//Set UDP streaming port
← STReam:CONNECTION:PORT?		//Check UDP streaming port
→ 4444		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

//Open/close connection and start/stop streaming		
← STReam:CONNECTION:OPEN		//Open connection and start streaming
← STReam:CONNECTION:STATE?		//Check connection state
→ CONNECTED		
← STReam:CONNECTION:CLOSE		//Close connection and stop streaming
← STReam:CONNECTION:STATE?		//Check connection state
→ CLOSED		
← SYSTem:ERRor:CODE:ALL?		//Check for errors
→ 0		//All OK

## Glossary / Acronyms

Term	Description
CBW:	Channel bandwidth
HTML	HTML stands for HyperText Markup Language and represents a language for describing Internet pages or their elements. Internet browsers are able to interpret this language and thus display the contents of the Internet pages formatted accordingly.
HTTP	HyperText Transfer Protocol is a standard, which describes the transfer of data over a network. It is mainly used to load / transfer web pages from the World Wide Web (WWW) into a web browser.
IP address	An IP address is a unique computer network address based on the Internet protocol (IP), which is assigned to each device connected to the network (e.g. the Internet) enabling the device to be addressed and therefore accessed (similar to the house number in a postal address).
Measurement mode (sub mode):	A second level of modes/a submenu of modes
Narda auto (directional) antenna:	A Narda directional antenna; which is always automatically detected by the device
Narda auto cable:	A Narda cable; which is always automatically detected by the device
NBW	Noise bandwidth
Option:	An additional measurement mode that can be enabled on the device by buying an (unlock-) code
RBW:	Resolution bandwidth
SCPI	Standard Commands for Programmable Instruments
TCP port	The TCP port number is part of the device network address. It indicates the application for which the data are intended. The TCP port number is thus comparable to the name of the recipient (addressee) in the analogy of a postal address.
User antenna:	An external antenna (non-Narda antenna)
User cable:	An external cable (non-Narda antenna)
VBW:	Video bandwidth





**Narda Safety Test Solutions GmbH**

Sandwiesenstrasse 7  
72793 Pfullingen, Germany  
Phone +49 7121 97 32 0  
info@narda-sts.com

**Narda Safety Test Solutions**

North America Representative Office  
435 Moreland Road  
Hauppauge, NY11788, USA  
Phone +1 631 231 1700  
info@narda-sts.com

**Narda Safety Test Solutions GmbH**

Beijing Representative Office  
Xiyuan Hotel, No. 1 Sanlihe Road, Haidian  
100044 Beijing, China  
Phone +86 10 6830 5870  
support@narda-sts.cn

[www.narda-sts.com](http://www.narda-sts.com)