



COPPER MOUNTAIN
TECHNOLOGIES

TR Series

Planar TR1300/1,
TR5048, TR7530

Vector Network Analyzer
Programming Manual SCPI



Software version: 19.3.0
August 2019

TABLE OF CONTENTS

1 Introduction	11
1.1 Programming Manual Scope	11
1.2 Related Documents.....	11
1.3 References.....	11
2 Analyzer Interfaces	12
2.1 Ethernet Interface	12
3 SCPI Overview.....	13
3.1 Command Tree	13
3.2 Subsystems	14
3.3 Optional Subsystems.....	14
3.4 Long and Short Formats	15
3.5 Case Sensitivity	15
3.6 Parameters	15
3.6.1 Numeric Values.....	15
3.6.1.1 Multiplier Prefixes.....	16
3.6.1.2 Notations	16
3.6.2 Booleans	17
3.6.3 Character Data.....	17
3.6.4 String Parameters	17
3.6.5 Numeric Lists.....	17
3.7 Query Commands	18
3.8 Numeric Suffixes.....	18
3.9 Compound Commands	18
3.10 IEEE488.2 Common Commands Overview.....	19
3.11 String Terminator.....	19
4 Remote Control Commands	20
4.1 Conventions.....	20
4.1.1 Syntax.....	20
4.1.2 Identifiers.....	21
4.2 IEEE488.2 Common Commands.....	22
*CLS	22
*ESE	23
*ESR?	23
*IDN?	24
*OPC	24
*OPC?	25
*RST	25
*SRE	26
*STB?	26
*TRG	27

*WAI	27
4.3 Network Analyzer Commands	28
ABOR	28
CALC:CONV	28
CALC:CONV:FUNC.....	29
CALC:CORR:EDEL:TIME	30
CALC:CORR:OFFS:PHAS	31
CALC:DATA:FDAT?.....	32
CALC:DATA:FMEM?	33
CALC:DATA:SDAT?	34
CALC:DATA:SMEM?.....	35
CALC:FILT:TIME.....	36
CALC:FILT:TIME:CENT	37
CALC:FILT:TIME:SHAP	38
CALC:FILT:TIME:SPAN	39
CALC:FILT:TIME:STAR.....	40
CALC:FILT:TIME:STAT	41
CALC:FILT:TIME:STOP.....	42
CALC:FORM.....	43
CALC:FSIM:SEND:DEEM:PORT:STAT	44
CALC:FSIM:SEND:DEEM:PORT:USER:FIL.....	45
CALC:FSIM:SEND:PMC:PORT:STAT	46
CALC:FSIM:SEND:PMC:PORT:USER:FIL	47
CALC:FSIM:SEND:ZCON:PORT:Z0.....	48
CALC:FSIM:SEND:ZCON:STAT	49
CALC:FUNC:DATA?	50
CALC:FUNC:DOM	51
CALC:FUNC:DOM:COUP.....	52
CALC:FUNC:DOM:STAR	53
CALC:FUNC:DOM:STOP	54
CALC:FUNC:EXEC.....	55
CALC:FUNC:PEXC.....	56
CALC:FUNC:POIN?	57
CALC:FUNC:PPOL	58
CALC:FUNC:TARG	59
CALC:FUNC:TTR	60
CALC:FUNC:TYPE	61
CALC:LIM	62
CALC:LIM:DATA.....	63
CALC:LIM:DISP.....	64
CALC:LIM:FAIL?	64
CALC:LIM:OFFS:AMPL.....	65
CALC:LIM:OFFS:STIM	66
CALC:LIM:REP:ALL?	67

CALC:LIM:REP:POIN?	68
CALC:LIM:REP?	69
CALC:MARK	70
CALC:MARK:ACT	71
CALC:MARK:BWID	71
CALC:MARK:BWID:DATA?	72
CALC:MARK:BWID:REF	73
CALC:MARK:BWID:THR	74
CALC:MARK:BWID:TYPE	75
CALC:MARK:COUN	76
CALC:MARK:COUP	76
CALC:MARK:FUNC:DOM	77
CALC:MARK:FUNC:DOM:STAR	78
CALC:MARK:FUNC:DOM:STOP	79
CALC:MARK:FUNC:EXEC	80
CALC:MARK:FUNC:PEXC	81
CALC:MARK:FUNC:PPOL	82
CALC:MARK:FUNC:TARG	83
CALC:MARK:FUNC:TRAC	84
CALC:MARK:FUNC:TTR	85
CALC:MARK:FUNC:TYPE	86
CALC:MARK:REF	87
CALC:MARK:SET	88
CALC:MARK:X	89
CALC:MARK:Y?	90
CALC:MATH:DEL	90
CALC:MATH:FUNC	91
CALC:MATH:MEM	92
CALC:MST	92
CALC:MST:DATA?	93
CALC:MST:DOM	94
CALC:MST:DOM:STAR	95
CALC:MST:DOM:STOP	96
CALC:PAR:COUN	97
CALC:PAR:DEF	98
CALC:PAR:SEL	99
CALC:PAR:SPOR	100
CALC:RLIM	101
CALC:RLIM:DATA	102
CALC:RLIM:DISP:LINE	103
CALC:RLIM:FAIL?	103
CALC:RLIM:REP?	104
CALC:SMO	105
CALC:SMO:APER	106

CALC:TRAC:DATA:FDAT?	107
CALC:TRAC:DATA:FMEM?	108
CALC:TRAC:DATA:SDAT?	109
CALC:TRAC:DATA:SMEM?	110
CALC:TRAN:TIME	111
CALC:TRAN:TIME:CENT	112
CALC:TRAN:TIME:IMP:WIDT	113
CALC:TRAN:TIME:KBES	114
CALC:TRAN:TIME:LPFR.....	114
CALC:TRAN:TIME:REFL:TYPE	115
CALC:TRAN:TIME:SPAN.....	116
CALC:TRAN:TIME:STAR	117
CALC:TRAN:TIME:STAT	118
CALC:TRAN:TIME:STEP:RTIM.....	119
CALC:TRAN:TIME:STIM.....	120
CALC:TRAN:TIME:STOP	121
CALC:TRAN:TIME:UNIT.....	122
DISP:COL:BACK	123
DISP:COL:GRAT	124
DISP:COL:RES.....	124
DISP:COL:TRAC:DATA	125
DISP:COL:TRAC:MEM	126
DISP:ENAB.....	126
DISP:FSIG.....	127
DISP:IMAG	127
SYST:HIDE	128
SYSTem:SHOW	128
DISP:MAX	129
DISP:SPL	129
DISP:UPD	130
DISP:WIND:ACT	131
DISP:WIND:ANN:MARK:ALIG	132
DISP:WIND:ANN:MARK:SING.....	133
DISP:WIND:MAX.....	133
DISP:WIND:SPL.....	134
DISP:WIND:TITL	135
DISP:WIND:TITL:DATA.....	135
DISP:WIND:TRAC:ANN:MARK:MEM.....	136
DISP:WIND:TRAC:ANN:MARK:POS:X	137
DISP:WIND:TRAC:ANN:MARK:POS:Y.....	138
DISP:WIND:TRAC:Y:AUTO.....	138
DISP:WIND:TRAC:Y:PDIV	139
DISP:WIND:TRAC:Y:RLEV	140
DISP:WIND:TRAC:Y:RPOS	141

DISP:WIND:Y:DIV	142
FORM:BORD	143
FORM:DATA.....	144
HCOP	145
HCOP:ABOR	145
HCOP:DATE:STAM.....	145
HCOP:IMAG.....	146
HCOP:PAIN.....	146
INIT	147
INIT:CONT	148
MMEM:COPY.....	148
MMEM:DEL.....	149
MMEM:LOAD	149
MMEM:LOAD:CHAN	150
MMEM:LOAD:CKIT.....	150
MMEM:LOAD:LIM.....	151
MMEM:LOAD:RLIM.....	152
MMEM:LOAD:SEGM	152
MMEM:MDIR.....	153
MMEM:STOR.....	153
MMEM:STOR:CHAN.....	154
MMEM:STOR:CHAN:CLE.....	154
MMEM:STOR:CKIT	155
MMEM:STOR:FDAT.....	155
MMEM:STOR:IMAG.....	156
MMEM:STOR:LIM.....	156
MMEM:STOR:RLIM.....	157
MMEM:STOR:SEGM.....	157
MMEM:STOR:SNP	158
MMEM:STOR:SNP:FORM.....	159
MMEM:STOR:SNP:TYPE:S1P.....	159
MMEM:STOR:SNP:TYPE:S2P	160
MMEM:STOR:STYP.....	161
MMEM:TRAN?	162
OUTP	162
SENS:AVER	163
SENS:AVER:CLE	163
SENS:AVER:COUN.....	164
SENS:BAND	165
SENS:BWID.....	165
SENS:CORR:CLE	166
SENS:CORR:COEF	167
SENS:CORR:COEF:METH:ERES	168
SENS:CORR:COEF:METH:OPEN	169

SENS:CORR:COEF:METH:SHOR	170
SENS:CORR:COEF:METH:SOLT1	171
SENS:CORR:COEF:METH:THRU	172
SENS:CORR:COEF:SAVE.....	173
SENS:CORR:COLL:CKIT.....	174
SENS:CORR:COLL:CKIT:LAB	175
SENS:CORR:COLL:CKIT:RES.....	176
SENS:CORR:COLL:CKIT:STAN:C0.....	176
SENS:CORR:COLL:CKIT:STAN:C1	177
SENS:CORR:COLL:CKIT:STAN:C2	178
SENS:CORR:COLL:CKIT:STAN:C3.....	179
SENS:CORR:COLL:CKIT:STAN:DEL	180
SENS:CORR:COLL:CKIT:STAN:L0	181
SENS:CORR:COLL:CKIT:STAN:L1	182
SENS:CORR:COLL:CKIT:STAN:L2	183
SENS:CORR:COLL:CKIT:STAN:L3	184
SENS:CORR:COLL:CKIT:STAN:LAB	185
SENS:CORR:COLL:CKIT:STAN:LOSS.....	186
SENS:CORR:COLL:CKIT:STAN:TYPE.....	187
SENS:CORR:COLL:CKIT:STAN:Z0	188
SENS:CORR:COLL:CLE	188
SENS:CORR:COLL:ECAL:CHECK:EXEC.....	189
SENS:CORR:COLL:ECAL:ORI:EXEC	189
SENS:CORR:COLL:ECAL:ORI:STAT.....	190
SENS:CORR:COLL:ECAL:PATH	191
SENS:CORR:COLL:ECAL:SOLT1	192
SENS:CORR:COLL:ECAL:SOLT2	192
SENS:CORR:COLL:ECAL:UCH.....	193
SENS:CORR:COLL:LOAD.....	194
SENS:CORR:COLL:METH:ERES.....	195
SENS:CORR:COLL:METH:OPEN	196
SENS:CORR:COLL:METH: SHOR	197
SENS:CORR:COLL:METH:SOLT1.....	198
SENS:CORR:COLL:METH:THRU	199
SENS:CORR:COLL:METH:TYPE?	200
SENS:CORR:COLL:OPEN.....	201
SENS:CORR:COLL:SAVE.....	202
SENS:CORR:COLL:SHOR	203
SENS:CORR:COLL:THRU	204
SENS:CORR:EXT	205
SENS:CORR:EXT:PORT:FREQ	206
SENS:CORR:EXT:PORT:INCL.....	207
SENS:CORR:EXT:PORT:LDC	208
SENS:CORR:EXT:PORT:LOSS.....	209

SENS:CORR:EXT:PORT:TIME	210
SENS:CORR:IMP	211
SENS:CORR:STAT	212
SENS:CORR:TRAN:TIME:FREQ	213
SENS:CORR:TRAN:TIME:LOSS.....	214
SENS:CORR:TRAN:TIME:RVEL.....	215
SENS:CORR:TRAN:TIME:STAT	216
SENS:CORR:TYPE?	217
SENS:FREQ.....	218
SENS:FREQ:DATA?	219
SENS:FREQ:CENT.....	220
SENS:FREQ:SPAN.....	221
SENS:FREQ:STAR	222
SENS:FREQ: STOP	223
SENS:OFFS	224
SENS:OFFS:ADJ.....	224
SENS:OFFS:ADJ:CONT	225
SENS:OFFS:ADJ:CONT:PER.....	226
SENS:OFFS:ADJ:EXEC	227
SENS:OFFS:ADJ:PORT	228
SENS:OFFS:ADJ:VAL.....	229
SENS:OFFS:PORT:DATA?	230
SENS:OFFS:PORT:DIV	231
SENS:OFFS:PORT:MULT.....	232
SENS:OFFS:PORT:OFFS.....	233
SENS:OFFS:PORT:STAR.....	234
SENS:OFFS:PORT:STOP.....	235
SENS:ROSC:SOUR.....	236
SENS:SEGM:DATA.....	237
SENS:SWE:POIN	238
SENS:SWE:POIN:TIME.....	239
SENS:SWE:TYPE	240
SENS:VVM:TYPE	241
SENS:VVM:FORM	242
SENS:VVM:FREQ.....	243
SENS:VVM:DATA?	244
SENS:VVM:REF:CLE	244
SENS:VVM:REF:MEM.....	245
SENS:VVM:REF:DATA?	245
SENS:VVM:TABL:CLE	246
SENS:VVM:TABL:MEM.....	246
SENS:VVM:TABL:INS.....	247
SENS:VVM:TABL:REM.....	247
SENS:VVM:TABL:SAVE	248

SENS:VVM:TABL:DATA?.....	249
SERV:CHAN:ACT?.....	250
SERV:CHAN:COUN?.....	250
SERV:CHAN:TRAC:ACT?.....	250
SERV:CHAN:TRAC:COUN?.....	251
SERV:PORT:COUN?.....	251
SERV:SWE:FREQ:MAX?.....	251
SERV:SWE:FREQ:MIN?.....	252
SERV:SWE:POIN?.....	252
SOUR:POW.....	253
SOUR:POW:CENT.....	254
SOUR:POW:SLOP.....	255
SOUR:POW:SPAN.....	256
SOUR:POW:STAR.....	257
SOUR:POW:STOP.....	258
STAT:OPER?.....	259
STAT:OPER:COND?.....	259
STAT:OPER:ENAB.....	260
STAT:OPER:NTR.....	261
STAT:OPER:PTR.....	262
STAT:PRES.....	262
STAT:QUES:COND?.....	263
STAT:QUES:ENAB.....	263
STAT:QUES:LIM:CHAN:COND?.....	264
STAT:QUES:LIM:CHAN:ENAB.....	264
STAT:QUES:LIM:CHAN:NTR.....	265
STAT:QUES:LIM:CHAN:PTR.....	266
STAT:QUES:LIM:CHAN?.....	267
STAT:QUES:LIM:COND?.....	267
STAT:QUES:LIM:ENAB.....	268
STAT:QUES:LIM:NTR.....	269
STAT:QUES:LIM:PTR.....	270
STAT:QUES:LIM?.....	270
STAT:QUES:NTR.....	271
STAT:QUES:PTR.....	272
STAT:QUES:RLIM:CHAN:COND?.....	273
STAT:QUES:RLIM:CHAN:ENAB.....	273
STAT:QUES:RLIM:CHAN:NTR.....	274
STAT:QUES:RLIM:CHAN:PTR.....	275
STAT:QUES:RLIM:CHAN?.....	275
STAT:QUES:RLIM:COND?.....	276
STAT:QUES:RLIM:ENAB.....	276
STAT:QUES:RLIM:NTR.....	277
STAT:QUES:RLIM:PTR.....	278

STAT:QUES:RLIM?	278
STAT:QUES?.....	279
SYST:COMM:ECAL:TEMP:SENS?	279
SYST:CORR.....	280
SYST:DATE	280
SYST:ERR?.....	281
SYST:LOC.....	281
SYST:PRES.....	282
SYST: REAdy?	282
SYST:REM	283
SYST:RWL	283
SYST:TEMP:SENS?	284
SYST:TIME	285
SYST:TERM.....	285
TRIG	286
TRIG:SING.....	287
TRIG:SOUR	288
TRIG:STAT?	289
TRIG:WAIT	290
VVM:RES.....	291
VVM:FONT	291
5 Programming Tips.....	292
5.1 Program Sweep Initiation and Waiting.....	292
5.2 External Trigger	293
5.3 Waiting for Calibration Commands	294
5.4 VISA Timeout Considerations	294
5.5 Receiving Data Arrays in Text Form	294
5.6 Receiving Data Arrays in Binary Form.....	296
Appendix 1. IEE488.2 Status Reporting System.....	297
Appendix 2. Error Codes	303
Appendix 3. Programming Examples	304

1 Introduction

This Manual contains information about the CMT Network Analyzer remote control and its data communication, carried out by means of user program through a computer network.

This manual describes the analyzer command system and the network protocols used to deliver commands. The commands sent to the analyzer and the responses from it are text messages conforming to the **SCPI** (Standard Commands for Programmable Instruments) specification. The text messages are delivered over computer networks using TCP/IP Socket network protocols.

TCP/IP Socket is a general-purpose network protocol. The user program can connect to the analyzer using the TCP/IP Socket protocol both directly and through the VISA library.

VISA (Virtual Instrument Software Architecture) is a widely used software input/output interface in the field of testing and measurement for controlling devices from a personal computer. It is a library of functions for C / C ++, C #, Visual Basic, MATLAB, LabVIEW and others. The VISA library unifies access to all measuring instruments, regardless of the protocol and equipment used. The VISA library is installed on the client side, that is, on the computer where the user program is executed. The VISA library is available on the websites of many companies for free.

1.1 Programming Manual Scope

This Manual directly applies to the Planar – TR1300/1, TR5048 and TR7530 models Network Analyzer.

1.2 Related Documents

Before reading this Manual, familiarize yourself with Analyzer Operating Manual.

1.3 References

IEEE Standard 488.2–1992, IEEE Standard Codes, Formats, Protocols and Common Commands for Use with ANSI/IEEE Std 488.1–1987. IEEE, New York, NY, 1992.

SCPI Standard–1999, Standard Commands for Programmable Instruments Volume 1: Syntax and Style. SCPI Consortium, San Diego, CA, 1999.

VISA specifications, <http://www.ivifoundation.org/specifications>.

2 Analyzer Interfaces

2.1 Ethernet Interface

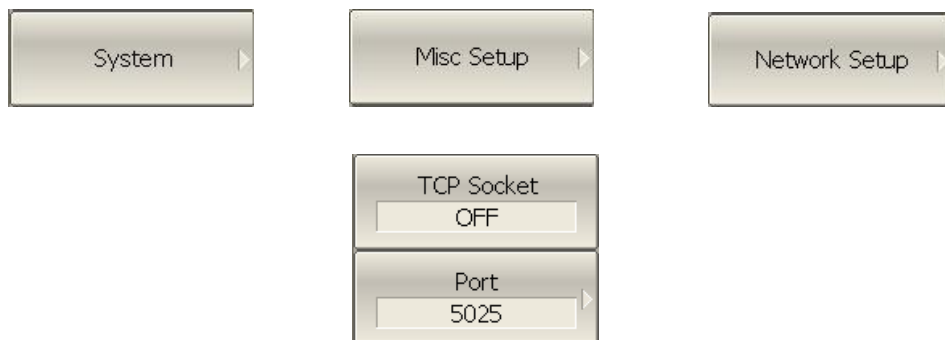
Ethernet interface is part of the equipment of a personal computer that connects to the device "Planar - TR1300/1», «TR5048» or «TR7530».

Data transfer between the PC user and the computer that is connected to the device, is performed via Socket protocol (TCP, port 5025).

Connect the device to a PC in the local Ethernet network of your company.

Activate the function of remote control via Socket protocol on your Analyzer as described below.

To enable/disable remote control via network Socket protocol, use the following softkeys: **System > Misc Setup > Network Setup > TCP Socket ON/OFF.**



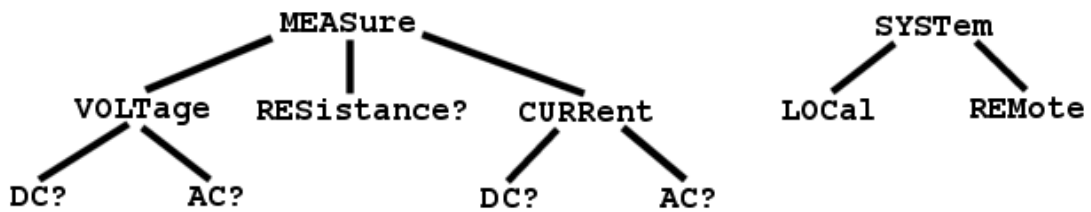
3 SCPI Overview

The Ethernet interface implement same set of commands based on 1999 SCPI standard (Standard Commands for Programmable Instruments). This command set allows the exchange of character messages.

SCPI is developed by SCPI Consortium (<http://www.scpiconsortium.org>).The main features of SCPI standard are described below. For a more detailed information on SCPI standard, see the website of SCPI Consortium.

3.1 Command Tree

The SCPI commands are organized in a tree structure. For example:



Each tree structure forms a functional system. The base of the tree is called root, e.g. MEASure and SYSTEM. Each functional system can have subsystems of lower level. The final nodes are called leaves. The entire sequence from root to the leaf makes up the command. For example, part of SOURCe functional system looks as follows:

```
:SOURce
    :POWer
        :CENTer
        :START
        :SPAN
        :STOP
        [:LEVel]
            :SLOPe
                [:DATA]
                :STATe
```

This `SOURCE` branch has several levels, where `CENTER`, `START`, `SPAN`, `STOP`, `DATA`, `STATE` are the leaves, which represent the following six commands:

```
:SOURCE:POWER:CENTER
:SOURCE:POWER:START
:SOURCE:POWER:SPAN
:SOURCE:POWER:STOP
:SOURCE:POWER[:LEVEL]:SLOPE[:DATA]
:SOURCE:POWER[:LEVEL]:SLOPE:STATE
```

The tree can contain subsystems and leaves with the same names if they belong to different branches, e.g. `CENTER` leaf is on the tips of different branches:

```
:SOURCE                                :SENSE
      :POWER                            :FREQUENCY
              :CENTER                    :CENTER
```

3.2 Subsystems

A colon (':') separates the subsystems. The subsystems, which follow the colon are of a lower level. For example, in command:

```
:SOURCE:POWER:START
```

the start power `START` is a part of `POWER` subsystem, which is a part of `SOURCE` subsystem. The stop power is also a part of `:SOURCE:POWER` subsystem. It is specified by:

```
:SOURCE:POWER:STOP
```

The first colon in the line can be omitted, for example:

```
SOURCE:POWER:STOP
```

3.3 Optional Subsystems

Some subsystems can be specified as optional, if omission of such a subsystem will not lead to ambiguity. This means that the subsystem can be omitted in command line. The optional subsystems are bracketed ("[]"). For example, if full command specification is written as:

```
SOURCE:POWER[:LEVEL]:SLOPE[:DATA]
```

subsystems `LEVEL` and `DATA` are optional. Therefore the both commands are valid:

```
SOURCE:POWER:LEVEL:SLOPE:DATA
```

```
SOURCE:POWER:SLOPE
```

3.4 Long and Short Formats

Each keyword in a command specification has a long format and a short format. The short format of a command is indicated by capital letters. For example, a command specification:

```
SENSe:FREQuency:CENTer
```

can be written as:

```
SENS:FREQ:CENT
```

```
SENS:FREQ:CENTer
```

Only long or short form of each keyword is acceptable. For example, the following specification is incorrect:

```
:SENS:FREQuen:CEn
```

3.5 Case Sensitivity

The commands are not case sensitive. Upper case and lower case letters are only used to indicate the long and short formats of a command specification. For example, the following commands are equivalent:

```
SENS:FREQ:STAR
```

```
sens:freq:star
```

3.6 Parameters

The commands can have parameters. The parameters are separated from the command by a space. If a command has several parameters, they are separated by commas (',').

3.6.1 Numeric Values

The numeric values are integer or real numbers. These parameters can have measurement units. For example:

```
SENS:FREQ 1000000000
```

```
SENS:FREQ 1000 MHz
```

```
SENS:FREQ 1 GHz
```

```
SENS:FREQ 1E9
```

3.6.1.1 Multiplier Prefixes

The SCPI standard allows specification of the numeric values with multiplier prefix to the measurement units.

Prefix	Multiplier
A	1e-18
F	1e-15
P	1e-12
N	1e-9
U	1e-6
M	1e-3
K	1e3
MA	1e6
G	1e9
T	1e12
PE	1e15
EX	1e18

There are two exceptions to the above designation: prefix M in combination with HZ or OHM means 1e6 (Mega), and not 1e-3 (milli), i.e. MHZ means Megahertz, same as MAHZ.

3.6.1.2 Notations

The SCPI standard allows numeric value specification in different notations. Decimal notation is used by default. To use other notations, specify the numeric values in the following way:

Notation	Prefix	Example
Binary	#B	#B11001010 = 202 ₁₀
Octal	#Q	#Q107 = 71 ₁₀
Hexadecimal	#H	#H10FF = 4351 ₁₀

3.6.2 Booleans

The booleans can assume two values: logical *yes* and logical *no* (ON and OFF), and specified in command as:

```
ON or 1 – logical yes
OFF or 0 – logical no
```

For example:

```
DISPlay:ENABle OFF
DISPlay:ENABle 0
```

3.6.3 Character Data

The SCPI standard allows specification of parameters as character data, as in the following command:

```
TRIGger:SOURce {BUS|IMMediate|EXTernal}
```

the possible values of the character data – "BUS", "IMMediate", "EXTernal".

The character data have long and short format, and the formats are specified in accordance with the same rules as described in Section 3.4.

Apart from that, the character data can be combined with numerical parameters. For example:

```
SENSe:FREQuency:STARt {MINimum|MAXimum|<value>}
```

The following specifications are acceptable:

```
SENSe:FREQuency:STARt MIN
SENSe:FREQuency:STARt maximum
SENSe:FREQuency:STARt 1000000
```

3.6.4 String Parameters

In some cases, the instrument can accept parameters made of character strings. Such strings are enclosed with single quotes (') or double quotes ("). For example, the file name in the state saving command:

```
MMEMory:STORe "state01.sta"
```

3.6.5 Numeric Lists

The numeric lists (<numeric list>) are used to specify a variable number of numerical parameters, for example:

```
CALC:LIMit:DATA 2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3
```

3.7 Query Commands

The query commands read out the parameter values from the instrument. After a query command has been sent, the response should return via remote control interface.

The query commands has a question mark ('?') in the end of the command. Many of the commands have two forms. The form with a question mark writes the parameter, the form without a question mark reads out the parameter. For example:

```
SENSe:FREQuency:STARt 1 MHz
SENSe:FREQuency:STARt?
```

3.8 Numeric Suffixes

The instrument contains several items of the same type, such as 9 channels, each of which in turn contains 8 traces, etc. A numeric suffix is used to denote the item number in a command. The suffix is added to the keyword of the item (channel, trace, etc). For example, in the following specification the channel number <Ch> and trace number <Tr> indicate the channel and trace, to which this command is addressed:

```
CALCulate<Ch>:PARAmeter<Tr>:DEFine
```

According to this specification, the command referred to the trace 2 of the channel 1 will be written as follows:

```
CALC1:PAR2:DEF
```

The numeric suffix can be omitted. In this case, it is 1 by default. For example, the following commands are equivalent:

```
CALC:PAR:DEF
CALC1:PAR1:DEF
```

3.9 Compound Commands

It is possible to enter more than one command in the same command line. The commands in the line are separated by a semicolon (;). The specification of the first command is valid for the following command, except for the last leaf before the semicolon. For example:

```
SENS:FREQ:STAR 1 MHZ;STOP 2MHZ
```

If you need to start the next command from the highest level of the structure, this command should start from a colon (':'):

```
SENS:FREQ:STAR 1 MHZ;:CALC:PAR:DEF S21
```

3.10 IEEE488.2 Common Commands Overview

A SCPI compatible instrument must support a set of common commands of IEEE488.2 standard. These commands start with an asterisk (*). The list of such commands see below:

*CLS
*ESE
*ESE?
*ESR?
*IDN?
*OPC
*OPC?
*RST
*SRE
*SRE?
*STB?
*TRG
*WAI

These commands are used for resetting, state queries, etc.

3.11 String Terminator

The <new line> character (ASCII 10) in the last data byte of the command string is used as a command terminator. The string of instructions sent to the instrument is executed after the instruction terminator is received.

4 Remote Control Commands

4.1 Conventions

The following conventions are used throughout the Manual.

4.1.1 Syntax

The following symbols are used in command syntax:

<>	identifiers enclosed in angular brackets indicated that a particular type of data must be specified
[]	part enclosed in square brackets can be omitted
{ }	part enclosed in curly brackets indicates that you must select one of the items in this part. Individual items are separated by a vertical bar " "
<i>Space</i>	space separates commands from parameters
,	comma separates adjacent parameters
...	ellipses indicates that parameters in that part are omitted

4.1.2 Identifiers

Identifier	Parameter	Description
<numeric>	Number	{<integer> <real>}
<frequency>	Frequency	<numeric>{ [HZ] KHZ MHZ GHZ }
<power>	Power	<numeric>{ [DBM] DBMW DBW KW W MW UW NW }
<time>	Time	<numeric>{ [S] MS US NS PS FS }
<logmag>	Log Amplitude	<numeric>[DB]
<phase>	Phase	<numeric>{ [DEG] MADEG KDEG MDEG UDEG }
<stimulus>	Stimulus	{<frequency> <power> <time>}
<response>	Response	{<logmag> <phase> <time>}
<numeric list>	Numeric List	<numeric 1>,<numeric 2>,...<numeric N>

4.2 IEEE488.2 Common Commands

*CLS

*CLS

<i>Description</i>	<p>Clears the following (no query)</p> <ul style="list-style-type: none"> • Error Queue • Status Byte Register • Standard Event Status Register • Operation Status Event Register • Questionable Status Event Register • Questionable Limit Status Event Register • Questionable Limit Channel Status Event Register
<i>Target</i>	Status Reporting System
<i>Equivalent Softkeys</i>	None

*ESE

*ESE <numeric>

*ESE?

<i>Description</i>	Sets or reads out the value of the Standard Event Status Enable Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> 0 to 255
<i>Out of Range</i>	Bitwise AND with 255 number
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

*ESR?

*ESR?

<i>Description</i>	Reads out the value of the Standard Event Status Register. Executing this command clears the register value (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

IDN?**IDN?**

<i>Description</i>	Reads out the instrument information string. The string format: <manufacturer>, <model>, <serial number>, <software version/firmware version> (query only). For example: Copper Mountain Technologies, Planar – TR1300/1, 00000101, 0.8.12/1.0
<i>Target</i>	Instrument
<i>Query Response</i>	String up to 40 characters
<i>Equivalent Softkeys</i>	None

OPC**OPC**

<i>Description</i>	Sets the OPC bit (bit 0) of the Standard Event Status Register at the completion of all pending operations (no query).
<i>Target</i>	Status Reporting System
<i>Equivalent Softkeys</i>	None

OPC?**OPC?**

<i>Description</i>	1 is read out at the completion of all pending operations (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	1
<i>Equivalent Softkeys</i>	None

RST**RST**

<i>Description</i>	Restores the default settings of the instrument (no query). There is difference from presetting the instrument with :SYST:PRES command – in this case the trigger mode is set to <i>Hold</i> .
<i>Target</i>	Instrument
<i>Related Commands</i>	SYSTem:PRESet
<i>Equivalent Softkeys</i>	None

***SRE**

*SRE <numeric>

*SRE?

<i>Description</i>	Sets or reads out the value of the Service Request Enable Register (command/query)
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> 0 to 255.
<i>Out of Range</i>	Bitwise AND with 255 number
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

***STB?**

*STB?

<i>Description</i>	Reads out the value of the Status Byte Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

TRG**TRG**

<i>Description</i>	<p>If the trigger source is set to LAN (set to BUS with the :TRIG:SOUR command), triggers a sweep.</p> <p>If the trigger source is not set to the bus or the instrument is not waiting for a trigger, the command is ignored.</p> <p>The command is completed before the end of the sweep. (no query)</p>
<i>Target</i>	Instrument
<i>Related Commands</i>	TRIG:SOUR
<i>Equivalent Softkeys</i>	None

WAI**WAI**

<i>Description</i>	<p>Waits for the execution of all commands sent before this command. (no query)</p>
<i>Target</i>	Instrument
<i>Equivalent Softkeys</i>	None

4.3 Network Analyzer Commands

ABOR

ABORt

<i>Description</i>	Aborts the sweep and switches the triggers of all the channels to <i>Hold</i> . The channels in <i>Continuous</i> trigger mode switch to waiting for a trigger. If the trigger source is set to Internal, the channel from waiting for a trigger turns to a new sweep. (no query)
<i>Equivalent Softkeys</i>	None

CALC:CONV

CALCulate<Ch>[:SElected]:CONVersion[:STATE] {ON|OFF|1|0}

CALCulate<Ch>[:SElected]:CONVersion[:STATE]?

<i>Description</i>	Sets or reads out the ON/OFF state of S-parameter conversion function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} S-parameter conversion function ON {OFF 0} S-parameter conversion function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Conversion > Conversion

CALC:CONV:FUNC

```
CALCulate<Ch>[:SElected]:CONVersion:FUNCTion <char>
```

```
CALCulate<Ch>[:SElected]:CONVersion:FUNCTion?
```

<i>Description</i>	Sets or reads out the S-parameter conversion function type (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies parameter:</p> <p>IMPedance : Equivalent transmission or reflection impedance, depending on the parameter S11 or S21</p> <p>ADMittance : Equivalent transmission or reflection conductance, depending on the parameter S11 or S21</p> <p>INVersion : Inverse S-parameter</p> <p>CONJugation : S-parameter conjugate</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ IMP ADM INV CONJ }
<i>Preset Value</i>	IMP
<i>Equivalent Softkeys</i>	Analysis > Conversion > Function { Impedance Z Admittance Y Inverse 1/S Conjugation }

CALC:CORR:EDEL:TIME

```
CALCulate<Ch>[:SElected]:CORRection:EDELay:TIME <time>
```

```
CALCulate<Ch>[:SElected]:CORRection:EDELay:TIME?
```

<i>Description</i>	Sets or reads out the value of the electrical delay (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the electrical delay value from -10 to 10
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Scale > Electrical Delay

CALC:CORR:OFFS:PHAS

CALCulate<Ch>[:SElected]:CORRection:OFFSet:PHASe <phase>

CALCulate<Ch>[:SElected]:CORRection:OFFSet:PHASe?

<i>Description</i>	Sets or reads out the value of the phase offset (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<phase> the phase offset value from -360 to 360
<i>Unit</i>	° (degree)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Scale > Phase Offset

CALC:DATA:FDAT?

CALCulate<Ch>[:SElected]:DATA:FDATa?

<i>Description</i>	<p>Reads out the formatted data array (query only). The formatted data array is the data, whose processing is completed including the formatting as the last step. Such data represent the data trace values as they are shown on the screen. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> real number in rectangular format, real part in polar and Smith chart formats;</p> <p><numeric 2n> 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	CALC:FORM FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:DATA:FMEM?

CALCulate<Ch>[:SElected]:DATA:FMEMory?

<i>Description</i>	<p>Reads out the formatted memory array (query only). The formatted memory array is the data, whose processing is completed including the formatting as the last step. Such data represent the memory trace values as they are shown on the screen. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> real number in rectangular format, real part in polar and Smith chart formats;</p> <p><numeric 2n> 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	If the memory is empty, an error occurs and the command is ignored.
<i>Related Commands</i>	<p>CALC:MATH:MEM</p> <p>CALC:FORM</p> <p>FORM:DATA</p>
<i>Equivalent Softkeys</i>	None

CALC:DATA:SDAT?

CALCulate<Ch>[:SElected]:DATA:SDATa?

<i>Description</i>	<p>Reads out the corrected data array (query only). The corrected data array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> the real part of corrected measurement; <numeric 2n> the imaginary part of corrected measurement.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:DATA:SMEM?

CALCulate<Ch>[:SElected]:DATA:SMEMory?

<i>Description</i>	<p>Reads out the corrected memory array (query only). The corrected memory array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> the real part of corrected measurement memory; <numeric 2n> the imaginary part of corrected measurement memory.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	If the memory is empty, an error occurs and the command is ignored.
<i>Related Commands</i>	CALC:MATH:MEM FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:FILT:TIME

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME[:TYPE] <char>
```

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME[:TYPE]?
```

<i>Description</i>	Sets or reads out the gate type of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> specifies the gate type: BPASs : Bandpass type NOTCh : Notch type
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{BPAS NOTC}
<i>Preset Value</i>	BPAS
<i>Equivalent Softkeys</i>	Analysis > Gating > Type

CALC:FILT:TIME:CENT

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:CENTer <time>
```

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:CENTer?
```

<i>Description</i>	Sets or reads out the gate center value of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the center value of the gate, the range varies depending on the frequency span and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Gating > Center

CALC:FILT:TIME:SHAP

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SHApe <char>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SHApe?

<i>Description</i>	Sets or reads out the gate shape of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the gate shape:</p> <p>MAXimum : Maximum shape</p> <p>WIDE : Wide shape</p> <p>NORMal : Normal shape</p> <p>MINimum : Minimum shape</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ MAX WIDE NORM MIN }
<i>Preset Value</i>	NORM
<i>Equivalent Softkeys</i>	Analysis > Gating > Shape > { Maximum Wide Normal Minimum }

CALC:FILT:TIME:SPAN

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SPAN <time>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SPAN?

<i>Description</i>	Sets or reads out the gate span value of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the span value of the gate, the range varies depending on the frequency span and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	2e-8
<i>Equivalent Softkeys</i>	Analysis > Gating > Span

CALC:FILT:TIME:STAR

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:START <time>
```

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:START?
```

<i>Description</i>	Sets or reads out the gate start value of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the start value of the gate, the range varies depending on the frequency span and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	-1e-8
<i>Equivalent Softkeys</i>	Analysis > Gating > Start

CALC:FILT:TIME:STAT

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STATe {ON|OFF|1|0}

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STATe?

<i>Description</i>	Sets or reads out the ON/OFF state of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Gating function ON {OFF 0} Gating function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Gating > Gating

CALC:FILT:TIME:STOP

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STOP <time>
```

```
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STOP?
```

<i>Description</i>	Sets or reads out the gate stop value of the gating function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the stop value of the gate, the range varies depending on the frequency span and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	+1e-8
<i>Equivalent Softkeys</i>	Analysis > Gating > Stop

CALC:FORM

```
CALCulate<Ch>[:SElected]:FORMat{<char>
```

```
CALCulate<Ch>[:SElected]:FORMat?
```

<i>Description</i>	Sets or reads out the trace format (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the trace format:</p> <p>MLOGarithmic : Logarithmic magnitude</p> <p>PHASe : Phase</p> <p>GDELay : Group delay time</p> <p>SLINear : Smith chart format (Lin)</p> <p>SLOGarithmic : Smith chart format (Log)</p> <p>SCOMplex : Smith chart format (Real/Imag)</p> <p>SMITH : Smith chart format (R + jX)</p> <p>SADMittance : Smith chart format (G + jB)</p> <p>PLINear : Polar format (Lin)</p> <p>PLOGarithmic : Polar format (Log)</p> <p>POLar : Polar format (Real/Imag)</p> <p>MLINear : Linear magnitude</p> <p>SWR : Voltage standing wave ratio</p> <p>REAL : Real part</p> <p>IMAGinary : Imaginary part</p> <p>UPHase : Expanded phase</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{MLOG PHAS GDEL SLIN SLOG SCOM SMIT SADM PLIN PLOG POL MLIN SWR REAL IMAG UPH }
<i>Preset Value</i>	MLOG
<i>Equivalent Softkeys</i>	<p>Response > Format > { Log Mag Phase Group Delay Lin Mag SWR Real Imag Phase Exp }</p> <p>Response > Format > Smith > { Log/Phase Lin/Phase Real/Imag R+jX G+jB }</p> <p>Response > Format > Polar > { Log/Phase Ling/Phase Real/Imag }</p>

CALC:FSIM:SEND:DEEM:PORT:STAT

```
CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:STATe
{ON|OFF|1|0}
```

```
CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the de-embedding function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	{ON 1} De-embedding function ON {OFF 0} De-embedding function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding > Port n

CALC:FSIM:SEND:DEEM:PORT:USER:FIL

```
CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:USER:FILEname
<string>
```

```
CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:USER:FILEname
?
```

<i>Description</i>	Sets or reads out the name of *.s2p file of the de-embedded circuit of the de-embedding function (command/query). The file contains the circuit S-parameters in Touchstone format.
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<string>, up to 256 characters (quoted string)
<i>Preset Value</i>	""
<i>Notes</i>	If the full path of the file is not specified, the \FixtureSim subdirectory of the main directory will be searched for the file.
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding > S-parameters File

CALC:FSIM:SEND:PMC:PORT:STAT

```
CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:STATe
{ON|OFF|1|0}
```

```
CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the embedding function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	{ON 1} Embedding function ON {OFF 0} Embedding function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Embedding > Port n

CALC:FSIM:SEND:PMC:PORT:USER:FIL

```
CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:USER:
FILename <string>
```

```
CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:USER:
FILename?
```

<i>Description</i>	Sets or reads out the name of *.s2p file of the embedded circuit of the embedding function (command/query). The file contains the circuit S-parameters in Touchstone format.
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<string>, up to 256 characters (quoted string)
<i>Preset Value</i>	""
<i>Notes</i>	If the full path of the file is not specified, the \FixtureSim subdirectory of the main directory will be searched for the file.
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Embedding > User File

CALC:FSIM:SEND:ZCON:PORT:Z0

```
CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0[:R]
<impedance>
```

```
CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0[:R]?
```

<i>Description</i>	Sets or reads out the value of the impedance for port impedance conversion function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<impedance> the impedance value from 1e-6 to 1e6
<i>Unit</i>	Ω (Ohm)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	50
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port n Z0

CALC:FSIM:SEND:ZCON:STAT

```
CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:STATe {ON|OFF|1|0}
```

```
CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the port impedance conversion function (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Port Z conversion function ON {OFF 0} Port Z conversion function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port Z Conversion

CALC:FUNC:DATA?

CALCulate<Ch>[:SElected]:FUNCTION:DATA?

<i>Description</i>	<p>Reads out the data array, which is the CALC:FUNC:EXEC command analysis result (query only). The array size is 2N, where N is the number of points set by CALC:FUNC:POIN? command. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> the response value in n-th measurement point;</p> <p><numeric 2n> the stimulus value in n-th measurement point. Always set to 0 for the analysis of mean value, standard deviation, and peak-to-peak value.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	<p>CALC:FUNC:EXEC CALC:FUNC:POIN? FORM:DATA</p>
<i>Equivalent Softkeys</i>	None

CALC:FUNC:DOM

CALCulate<Ch>[:SELEcted]:FUNction:DOMain[:STATe] {ON|OFF|1|0}

CALCulate<Ch>[:SELEcted]:FUNction:DOMain[:STATe]?

<i>Description</i>	Sets or reads out the ON/OFF state of an arbitrary range use in CALC:FUNC:EXEC command (command/query).
<i>Target</i>	All traces of channel <Ch> (if the coupling is set to ON by the CALC:FUNC:DOM:COUP command), the active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Arbitrary range ON {OFF 0} Arbitrary range OFF (entire sweep range)
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Related Commands</i>	CALC:FUNC:EXEC CALC:FUNC:DOM:COUP
<i>Equivalent Softkeys</i>	None

CALC:FUNC:DOM:COUP

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:COUPle {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:COUPle?
```

<i>Description</i>	Sets or reads out the ON/OFF state of trace coupling for analysis range of the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	All traces of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} coupling ON {OFF 0} coupling OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:FUNC:DOM:STAR

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:STARt <stimulus>
```

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:STARt?
```

<i>Description</i>	Sets the start value of the analysis range of the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	All traces of channel <Ch> (if the coupling is set to ON by the CALC:FUNC:DOM:COUP command), the active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<stimulus> the start value of analysis range
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Related Commands</i>	CALC:FUNC:DOM
<i>Equivalent Softkeys</i>	None

CALC:FUNC:DOM:STOP

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:STOP <stimulus>
```

```
CALCulate<Ch>[:SElected]:FUNction:DOMain:STOP?
```

<i>Description</i>	Sets the stop value of the analysis range of the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	All traces of channel <Ch> (if the coupling is set to ON by the CALC:FUNC:DOM:COUP command), the active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<stimulus> the stop value of analysis range
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Related Commands</i>	CALC:FUNC:DOM
<i>Equivalent Softkeys</i>	None

CALC:FUNC:EXEC

CALCulate<Ch>[:SElected]:FUNction:EXECute

<i>Description</i>	Executes the analysis specified by the CALC:FUNC:TYPE command. The analysis result can be read out by the CALC:FUNC:DATA? command. (no query)
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Related Commands</i>	CALC:FUNC:TYPE CALC:FUNC:DATA?
<i>Equivalent Softkeys</i>	None

CALC:FUNC:PEXC

```
CALCulate<Ch>[:SElected]:FUNction:PEXCursion <response>
```

```
CALCulate<Ch>[:SElected]:FUNction:PEXCursion?
```

<i>Description</i>	Sets the lower limit for the peak excursion value when executing the peak search with the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<response> the lower limit of the peak excursion value, varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	3
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:FUNC:POIN?

CALCulate<Ch>[:SElected]:FUNCtion:POINTs?

<i>Description</i>	<p>Reads out the number of points (data pairs) of the analysis result by the CALC:FUNC:EXEC command (query only). Always reads out 1, when the search is executed for the maximum, minimum, mean, standard deviation, peak, and peak-to-peak values. The actual number of points is read out, when the search is executed for all peak or all targets.</p>
<i>Target</i>	<p>The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<p><numeric></p>
<i>Related Commands</i>	<p>CALC:FUNC:EXEC</p>
<i>Equivalent Softkeys</i>	<p>None</p>

CALC:FUNC:PPOL

```
CALCulate<Ch>[:SElected]:FUNction:PPOLarity <char>
```

```
CALCulate<Ch>[:SElected]:FUNction:PPOLarity?
```

<i>Description</i>	Selects the polarity when performing the peak search with the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the polarity:</p> <p>POSitive : Positive peaks</p> <p>NEGative : Negative peaks</p> <p>BOTH : Both positive peaks and negative peaks</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ POS NEG BOTH }
<i>Preset Value</i>	POS
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:FUNC:TARG

```
CALCulate<Ch>[:SElected]:FUNction:TARGet <response>
```

```
CALCulate<Ch>[:SElected]:FUNction:TARGet?
```

<i>Description</i>	Selects the target level when performing the search for the trace and the target level crosspoints with the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<response> the target value, varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:FUNC:TTR

```
CALCulate<Ch>[:SElected]:FUNction:TTRansition <char>
```

```
CALCulate<Ch>[:SElected]:FUNction:TTRansition?
```

<i>Description</i>	Selects the transition type when performing the search for the trace and the target level crosspoints with the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the type of transition:</p> <p>POSitive : Positive peaks</p> <p>NEGative : Negative peaks</p> <p>BOTH : Both positive peaks and negative peaks</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ POS NEG BOTH }
<i>Preset Value</i>	POS
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:FUNC:TYPE

```
CALCulate<Ch>[:SElected]:FUNction:TYPE <char>
```

```
CALCulate<Ch>[:SElected]:FUNction:TYPE?
```

<i>Description</i>	Selects the type of analysis executed with the CALC:FUNC:EXEC command (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the transition:</p> <p>PTPeak : Peak-to-peak (difference between the maximum value and the minimum value)</p> <p>STDEV : Standard deviation</p> <p>MEAN : Mean value</p> <p>MAXimum : Maximum value</p> <p>MINimum : Minimum value</p> <p>PEAK : Search for peak</p> <p>APEak : Search for all the peaks</p> <p>ATARget : Search for all targets</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ PTP STDEV MEAN MAX MIN PEAK APE ATAR }
<i>Preset Value</i>	PTP
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	None

CALC:LIM

```
CALCulate<Ch>[:SElected]:LIMit[:STATe] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:LIMit[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the limit test function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Limit test function ON {OFF 0} Limit test function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Test

CALC:LIM:DATA

```
CALCulate<Ch>[:SElected]:LIMit:DATA <numeric list>
```

```
CALCulate<Ch>[:SElected]:LIMit:DATA?
```

<i>Description</i>	<p>Sets the data array, which is the limit line in the limit test function (command/query). The array size is $1 + 5N$, where N is the number of limit line segments. For the n-th point, where n from 1 to N:</p> <p><code><numeric 1></code> the number of limit line segments N is from 0 to 100. Setting 0 clears the limit line.</p> <p><code><numeric 5n -3></code> type of the n-th limit line segment</p> <ul style="list-style-type: none"> • 0: Off. • 1: Upper limit • 2: Lower limit • 3: Single Point limit <p><code><numeric 5n-2></code> the stimulus value in the start point of the n-th segment</p> <p><code><numeric 5n-1></code> the stimulus value in the end point of the n-th segment</p> <p><code><numeric 5n-0></code> the response value in the start point of the n-th segment</p> <p><code><numeric 5n+1></code> the response value in the end point of the n-th segment</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><code><numeric 1></code>, <code><numeric 2></code>, ...<code><numeric 5N+1></code></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	If the array size is not $1 + 5N$, where N is <code><numeric 1></code> , an error occurs. If <code><numeric 5n-3></code> is less than 0 or more than 2, an error occurs. When <code><numeric 5n-2></code> , <code><numeric 5n-1></code> , <code><numeric 5n-0></code> , and <code><numeric 5n+1></code> elements are out of allowable range, the value is set to the limit, which is closer to the specified value.
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line

CALC:LIM:DISP

```
CALCulate<Ch>[:SElected]:LIMit:DISPlay[:STATe] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:LIMit:DISPlay[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the limit line display of the limit test function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Limit line display ON {OFF 0} Limit line display OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line

CALC:LIM:FAIL?

```
CALCulate<Ch>[:SElected]:LIMit:FAIL?
```

<i>Description</i>	Reads out the limit test result (query only).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	1 Fail 0 Pass
<i>Equivalent Softkeys</i>	None

CALC:LIM:OFFS:AMPL

```
CALCulate<Ch>[:SElected]:LIMit:OFFSet:AMPLitude <response>
```

```
CALCulate<Ch>[:SElected]:LIMit:OFFSet:AMPLitude?
```

<i>Description</i>	Sets the value of the limit line offset along Y-axis (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<response> the value of the limit line offset along Y-axis, varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line Offsets > Response Offset

CALC:LIM:OFFS:STIM

```
CALCulate<Ch>[:SElected]:LIMit:OFFSet:STIMulus <stimulus>
```

```
CALCulate<Ch>[:SElected]:LIMit:OFFSet:STIMulus?
```

<i>Description</i>	Sets the value of the limit line offset along X-axis (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<stimulus> the value of the limit line offset along X-axis
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Lines Offsets > Stimulus Offset

CALC:LIM:REP:ALL?

CALCulate<Ch>[:SElected]:LIMit:REPort:ALL?

<i>Description</i>	<p>Reads out the data array, which is the limit test results (query only). The array size is 4N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 4n-3> the stimulus value in the n-th point;</p> <p><numeric 4n-2> the limit test result in the n-th point;</p> <ul style="list-style-type: none"> • -1: No limit • 0: Fail • 1: Pass <p><numeric 4n-1> the upper limit value in the n-th point (0 – if there is no limit)</p> <p><numeric 4n-0> the lower limit value in the n-th point (0 – if there is no limit)</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 4N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:LIM:REP:POIN?

CALCulate<Ch>[:SElected]:LIMit:REPort:POINts?

<i>Description</i>	Reads out the number of the measurement points that failed the limit test (query only). The stimulus data array of these points can be read out by the CALC:LIM:REP? command.
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric>
<i>Related Commands</i>	CALC:LIM:REP?
<i>Equivalent Softkeys</i>	None

CALC:LIM:REP?

CALCulate<Ch>[:SElected]:LIMit:REPort[:DATA]?

<i>Description</i>	Reads out the data array, which is the stimulus values of the measurement points that failed the limit test (query only). The array size is set by the CALC:LIM:REP:POIN? command.
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric N> The data transfer format depends on the FORM:DATA command setting.
<i>Related Commands</i>	CALC:LIM:REP:POIN? FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:MARK

```
CALCulate<Ch>[:SElected]:MARKer<Mk>[:STATe] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of a marker (command/query). Turning ON a marker with the number from 1 to 15 will turn ON all the markers of smaller numbers. Turning OFF a marker with the number from 1 to 15 will turn OFF all the markers of greater numbers (except of the reference marker). Turning ON/OFF the reference marker with number 16 does not turn ON/OFF the markers with the numbers from 1 to 15, but switches these markers to the relative measurement mode.
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	{ON 1} marker ON {OFF 0} marker OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Add Marker Remove Marker Markers > Reference Marker

CALC:MARK:ACT

CALCulate<Ch>:SElected]:MARKer<Mk>:ACTivate

<i>Description</i>	<p>Sets the active marker (no query).</p> <p>If the marker is not ON, this function will turn the marker ON. Turning ON a marker with the number from 1 to 15 will turn ON all the markers of smaller numbers. Turning ON the reference marker with number 16 does not turn ON the markers with the numbers from 1 to 15, but switches these markers to the relative measurement mode.</p>
<i>Target</i>	<p>Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }</p>
<i>Equivalent Softkeys</i>	<p>Markers > Active Marker > Marker n Markers > Reference Marker</p>

CALC:MARK:BWID

CALCulate<Ch>[:SElected]:MARKer:BWIDth[:STATe] {ON|OFF|1|0}

CALCulate<Ch>[:SElected]:MARKer:BWIDth[:STATe]?

<i>Description</i>	<p>Sets or reads out the ON/OFF state of the bandwidth search function (command/query).</p>
<i>Target</i>	<p>The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Parameter</i>	<p>{ON 1} Bandwidth search function ON {OFF 0} Bandwidth search function OFF</p>
<i>Query Response</i>	<p>{0 1}</p>
<i>Preset Value</i>	<p>0</p>
<i>Equivalent Softkeys</i>	<p>Markers > Marker Math > Bandwidth Search > Bandwidth Search</p>

CALC:MARK:BWID:DATA?

CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:DATA?

<i>Description</i>	<p>Reads out the bandwidth search result (query only). The bandwidth search can performed relatively to the marker <Mk>, or relatively to the absolute maximum value of the trace (in this case the number of the marker is ignored), what is set by the CALC:MARK:BWID:REF command. The data include 4 elements:</p> <p><numeric 1> Bandwidth; <numeric 2> Center frequency; <numeric 3> Q value; <numeric 4> Loss;</p>
<i>Target</i>	<p>Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9} <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16}</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric 4>
<i>Related Commands</i>	CALC:MARK:BWID:REF
<i>Notes</i>	<p>If the bandwidth search is impossible, all the read out values are 0. If the search is performed relatively to a maker, which is OFF, an error occurs.</p>
<i>Equivalent Softkeys</i>	None

CALC:MARK:BWID:REF

```
CALCulate<Ch>[:SElected]:MARKer:BWIDth:REFerence <char>
```

```
CALCulate<Ch>[:SElected]:MARKer:BWIDth:REFerence?
```

<i>Description</i>	Selects the reference point for the bandwidth search function: reference marker or absolute maximum value of the trace (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> choose from:</p> <p>MARKer : Bandwidth search relative to the reference marker</p> <p>MAXimum : Bandwidth search relative to the absolute maximum of the trace</p> <p>MINimum : Bandwidth search relative to the absolute minimum of the trace</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{MAX MARK MIN}
<i>Preset Value</i>	MAX
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Search Ref To

CALC:MARK:BWID:THR

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:THReshold
<response>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:THReshold?
```

<i>Description</i>	Sets the bandwidth definition value. (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<response> the bandwidth definition value, the range varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	-3
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Bandwidth Value

CALC:MARK:BWID:TYPE

```
CALCulate<Ch>[:SElected]:MARKer:BWIDth:TYPE <char>
```

```
CALCulate<Ch>[:SElected]:MARKer:BWIDth:TYPE?
```

<i>Description</i>	Sets the type of the bandwidth search function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> specifies the type of the bandwidth: BPASs : Bandpass NOTCh : Notch
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{BPAS NOTC}
<i>Preset Value</i>	BPAS
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Type

CALC:MARK:COUN

```
CALCulate<Ch>[:SElected]:MARKer:COUNT <numeric>
```

```
CALCulate<Ch>[:SElected]:MARKer:COUNT?
```

<i>Description</i>	Sets the number of the turned ON markers (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric>, range from 0 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

CALC:MARK:COUP

```
CALCulate<Ch>[:SElected]:MARKer:COUPLE {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MARKer:COUPLE?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the marker coupling function (command/query).
<i>Target</i>	All the traces of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Marker coupling ON {OFF 0} Marker coupling OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Marker > Properties > Marker Couple

CALC:MARK:FUNC:DOM

```
CALCulate<Ch>[:SElected]:MARKer:FUNc:DOMain[:STATe]
{ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MARKer:FUNc:DOMain[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the marker search range (command/query).
<i>Target</i>	All traces of channel <Ch> (if the marker search range coupling is set to ON by the CALC:MARK:FUNC:DOM:COUP command), The active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Marker search range ON {OFF 0} Marker search range OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Range

CALC:MARK:FUNC:DOM:STAR

```
CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STARt
<stimulus>
```

```
CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STARt?
```

<i>Description</i>	Sets or reads out the start value of the marker search range (command/query).
<i>Target</i>	All traces of channel <Ch> (if the marker search range coupling is set to ON by the CALC:MARK:FUNC:DOM:COUP command), The active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<stimulus> the start value of the marker search
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	3E5 (Planar – TR1300/1) 2E4 (TR5048) 2E4 (TR7530)
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Start

CALC:MARK:FUNC:DOM:STOP

```
CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STOP
<stimulus>
```

```
CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STOP?
```

<i>Description</i>	Sets or reads out the stop value of the marker search range (command/query).
<i>Target</i>	All traces of channel <Ch> (if the marker search range coupling is set to ON by the CALC:MARK:FUNC:DOM:COUP command), The active trace of channel <Ch> (if otherwise), <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<stimulus> the stop value of the marker search
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1.3E9 (Planar – TR1300/1) 4.8E9 (TR5048) 3.0E9 (TR7530)
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Stop

CALC:MARK:FUNC:EXEC

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:EXECute

<i>Description</i>	Executes the marker search according to the specified criterion. The type of the marker search is set by the CALC:MARK:FUNC:TYPE command. (no query)
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Related Commands</i>	CALC:MARK:FUNC:TYPE CALC:MARK:FUNC:DOM
<i>Equivalent Softkeys</i>	Markers > Marker Search > { Maximum Minimum } Markers > Marker Search > Peak > { Search Peak Search Max Peak Search Peak Left Search Peak Right } Markers > Marker Search > Target > { Search Target Search Target Left Search Target Right }

CALC:MARK:FUNC:PEXC

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:PEXCursion
<response>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:PEXCursion?
```

<i>Description</i>	Sets or reads out the peak excursion value, when the marker search for peak is performed by the CALC:MARK:FUNC:EXEC command (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<response> the peak excursion value, the range varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Excursion

CALC:MARK:FUNC:PPOL

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:PPOLarity <char>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:PPOLarity?
```

<i>Description</i>	Selects the peak polarity, when the marker search for peak is performed by the CALC:MARK:FUNC:EXEC command (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<char> specifies the peak polarity: POSitive : Positive polarity NEGative : Negative polarity BOTH : Both positive polarity and negative polarity
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ POS NEG BOTH }
<i>Preset Value</i>	POS
<i>Related Commands</i>	CALC:MARK:FUNC:EXEC
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Polarity > { Positive Negative Both }

CALC:MARK:FUNC:TARG

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:TARGet <response>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:TARGet?
```

<i>Description</i>	Sets or reads out the target value, when the marker search for target is performed by the CALC:MARK:FUNC:EXEC command (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<response> the target value, the range varies depending on the data format
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Marker Search > Target > Target Value

CALC:MARK:FUNC:TRAC

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:TRACking
{ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNction:TRACking?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the marker search tracking function (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	{ON 1} Marker search tracking ON {OFF 0} Marker search tracking OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Marker Search > Tracking

CALC:MARK:FUNC:TTR

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TTRansition
<char>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TTRansition?
```

<i>Description</i>	Selects the type of the target transition, when the marker search for transition is performed by the CALC:MARK:FUNC:EXEC command (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<char> specifies the type of the target transition: POSitive : Positive target transition NEGative : Negative target transition BOTH : Both positive target transition and negative target transition
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ POS NEG BOTH }
<i>Preset Value</i>	POS
<i>Related Commands</i>	CALC:MARK:FUNC:EXEC
<i>Equivalent Softkeys</i>	Marker > Marker Search > Target > Target Transition > { Positive Negative Both }

CALC:MARK:FUNC:TYPE

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TYPE <char>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TYPE?
```

<i>Description</i>	Selects the type of the marker search, which is performed by the CALC:MARK:FUNC:EXEC command (command/query).
<i>Target</i>	Marker <Mk> the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<char> specifies the type of the marker search: MAXimum : Maximum value search MINimum : Minimum value search PEAK : Peak search LPEak : Peak search to the left from the marker RPEak : Peak search to the right from the marker TARGet : Target search LTARget : Target search to the left from the marker RTARget : Target search to the right from the marker
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ MAX MIN PEAK LPE RPE TARG LTAR RTAR }
<i>Preset Value</i>	MAX
<i>Related Commands</i>	CALC:MARK:FUNC:EXEC
<i>Equivalent Softkeys</i>	Markers > Marker Search > { Maximum Minimum } Markers > Marker Search > Peak > { Search Peak Search Max Peak Search Peak Left Search Peak Right } Markers > Marker Search > Target > { Search Target Search Target Left Search Target Right }

CALC:MARK:REF

CALCulate<Ch>[:SElected]:MARKer:REFerence[:STATe] {ON|OFF|1|0}

CALCulate<Ch>[:SElected]:MARKer:REFerence[:STATe]?

<i>Description</i>	Sets or reads out the ON/OFF state of the reference marker (command/query). When the reference marker is turned ON, all the values of the other markers turn to relative values.
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Reference marker ON {OFF 0} Reference marker OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Reference Marker

CALC:MARK:SET

CALCulate<Ch>[:SElected]:MARKer<Mk>:SET <char>

<i>Description</i>	Sets the value of the specified item to the value of the position of the marker (no query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<p><char> specifies the type of the marker search:</p> <p>START : Sweep start value set to the stimulus value of the marker position.</p> <p>STOP : Sweep stop value set to the stimulus value of the marker position.</p> <p>CENTER : Sweep center value set to the stimulus value of the marker position.</p> <p>RLEVEL : Reference value set to the response value of the marker position.</p>
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	Markers > Marker Functions > { Marker->Start Marker->Stop Marker->Center Marker->Ref Value }

CALC:MARK:X

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:X <stimulus>
```

```
CALCulate<Ch>[:SElected]:MARKer<Mk>:X?
```

<i>Description</i>	Sets or reads out the stimulus value of the marker (command/query).
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Parameter</i>	<stimulus> the stimulus value of the marker, the range is from the stimulus start value to the stimulus stop value currently set
<i>Unit</i>	{ Hz (Hertz) s (second) dBm (decibels above 1 milliwatt) }
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	Stimulus center value
<i>Equivalent Softkeys</i>	Markers > Edit Stimulus

CALC:MARK:Y?

CALCulate<Ch>[:SElected]:MARKer<Mk>:Y?

<i>Description</i>	<p>Reads out the response value of the marker (query only). If the reference marker is turned ON, the values of the markers from 1 to 15 are read out as relative values to the reference marker. The data include 2 elements:</p> <p><numeric 1> real number in rectangular format, real part in polar and Smith chart formats;</p> <p><numeric 2> 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9} <Mk>={ [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16}</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>
<i>Related Commands</i>	CALC:MARK:REF
<i>Equivalent Softkeys</i>	None

CALC:MATH:DEL

CALCulate<Ch>[:SElected]:MATH:DElete

<i>Description</i>	Removes the data (All memory traces) from the memory trace.
<i>Target</i>	<p>The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}</p>
<i>Equivalent Softkeys</i>	Trace > Delete All Memory

CALC:MATH:FUNC

CALCulate<Ch>[:SElected]:MATH:FUNCTION <char>

CALCulate<Ch>[:SElected]:MATH:FUNCTION?

<i>Description</i>	Selects the math operation between the measurement data and the memory trace data. The math result replaces the data trace. If the data trace is not saved, the command is ignored (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the math operation:</p> <p>DIVide : Division Data / Mem.</p> <p>MULTIply : Multiplication Data x Mem.</p> <p>ADD : Addition Data + Mem.</p> <p>SUBTract : Subtraction Data – Mem.</p> <p>NORMal : No math</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ NORM SUBT DIV ADD MULT }
<i>Preset Value</i>	NORM
<i>Related Commands</i>	CALC:MATH:MEM
<i>Equivalent Softkeys</i>	Trace > Data Math > { Data/Mem Data*Mem Data+Mem Data–Mem OFF }

CALC:MATH:MEM

```
CALCulate<Ch>[:SElected]:MATH:MEMorize
```

<i>Description</i>	Saves the measurement data to the memory trace. Automatically turns on the display the memory trace (no query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	Trace > Display > Data->Memory

CALC:MST

```
CALCulate<Ch>[:SElected]:MSTatistics[:STATE] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MSTatistics[:STATE]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the math statistics display (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Statistics display ON {OFF 0} Statistics display OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics

CALC:MST:DATA?

CALCulate<Ch>[:SElected]:MStatistics:DATA?

<i>Description</i>	<p>Reads out the math statistics values (query only). The statistics function is applied either over the whole range (for all the trace), or within the range specified by CALC:MST:DOM command (the range limits are determined by two markers). The data include 3 elements:</p> <p><numeric 1> Mean value;</p> <p><numeric 2> Standard deviation;</p> <p><numeric 3> Peak-to-peak (difference between the maximum value and the minimum value).</p>
<i>Target</i>	<p>The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Related Commands</i>	CALC:MST:DOM
<i>Equivalent Softkeys</i>	None

CALC:MST:DOM

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:STATe]
{ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the math statistics range (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Statistics range ON {OFF 0} Statistics range OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Range

CALC:MST:DOM:STAR

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:MARKer]:START
<numeric>
```

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:MARKer]:START?
```

<i>Description</i>	Sets or reads out the number of the marker, which specifies the start frequency of the math statistics range (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> marker number from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Start

CALC:MST:DOM:STOP

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:MARKer]:STOP
<numeric>
```

```
CALCulate<Ch>[:SElected]:MStatistics:DOMain[:MARKer]:STOP?
```

<i>Description</i>	Sets or reads out the number of the marker, which specifies the stop frequency of the math statistics range (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> marker number from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	2
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Stop

CALC:PAR:COUN

```
CALCulate<Ch>:PARAmeter:COUNT <numeric>
```

```
CALCulate<Ch>:PARAmeter:COUNT?
```

<i>Description</i>	Sets or reads out the number of traces in the channel (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the number of the trace in the channel from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	None

CALC:PAR:DEF

```
CALCulate<Ch>:PARAmeter<Tr>:DEFine <char>
```

```
CALCulate<Ch>:PARAmeter<Tr>:DEFine?
```

<i>Description</i>	Sets the measurement parameter of the trace (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> specifies the measurement parameter: S11 : 11 parameter S21 : S21 parameter A : Receiver A B : Receiver B R : Receiver R
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	{ S11 S21 A B R }
<i>Preset Value</i>	Depends on the trace number. Tr 1, Tr 5, Tr 9, Tr 13 "S11" Tr 2, Tr 6, Tr 10, Tr 14 "S21" Tr 3, Tr 7, Tr 11, Tr 15 "S11" Tr 4, Tr 8, Tr 12, Tr 16 "S21"
<i>Equivalent Softkeys</i>	Response > Measurement > S11 S21 Response > Measurement > { Abs A Abs B Abs R }

CALC:PAR:SEL

CALCulate<Ch>:PARAmeter<Tr>:SElect

<i>Description</i>	Sets the active trace in channel <Ch> (no query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Notes</i>	If the trace number is greater than the number of the traces displayed in the channel, an error occurs and the command is ignored.
<i>Related Commands</i>	CALC:PAR:COUN
<i>Equivalent Softkeys</i>	Trace > Active Trace

CALC:PAR:SPOR

```
CALCulate<Ch>:PARAmeter<Tr>:SPORT <port>
```

```
CALCulate<Ch>:PARAmeter<Tr>:SPORT?
```

<i>Description</i>	Sets or reads out the number of the receiver port, when performing absolute measurements (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the receiver port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Response > Measurement > { Abs A Abs B }

CALC:RLIM

```
CALCulate<Ch>[:SElected]:RLIMit[:STATe] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:RLIMit[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the ripple limit test (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Ripple limit test ON {OFF 0} Ripple limit test OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Test

CALC:RLIM:DATA

```
CALCulate<Ch>[:SElected]:RLIMit:DATA <numeric list>
```

```
CALCulate<Ch>[:SElected]:RLIMit:DATA?
```

<i>Description</i>	<p>Sets the data array, which is the limit line for the ripple limit function (command/query). The array size is $1 + 4N$, where N is the number of limit line segments. For the n-th point, where n from 1 to N:</p> <p>$\langle \text{numeric } 1 \rangle$ the number of limit line segments N is the integer from 0 to 12. Setting 0 clears the limit line.</p> <p>$\langle \text{numeric } 4n - 2 \rangle$ type of the n-th limit line segment</p> <ul style="list-style-type: none"> • 0: Off. • 1: On <p>$\langle \text{numeric } 4n - 1 \rangle$ the stimulus value in the beginning point of the n-th segment</p> <p>$\langle \text{numeric } 4n - 0 \rangle$ the stimulus value in the end point of the n-th segment</p> <p>$\langle \text{numeric } 4n + 1 \rangle$ the ripple limit value of the n-th segment</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<p>$\langle \text{numeric } 1 \rangle$, $\langle \text{numeric } 2 \rangle$, ...$\langle \text{numeric } 4N + 1 \rangle$</p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	If the array size is not $1 + 4N$, where N is $\langle \text{numeric } 1 \rangle$, an error occurs. If $\langle \text{numeric } 4n - 2 \rangle$ is less than 0 or more than 1, an error occurs. When $\langle \text{numeric } 4n - 1 \rangle$, $\langle \text{numeric } 4n - 0 \rangle$, and $\langle \text{numeric } 4n + 1 \rangle$ elements are out of allowable range, the value is set to the limit, which is closer to the specified value.
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit

CALC:RLIM:DISP:LINE

```
CALCulate<Ch>[:SElected]:RLIMit:DISPlay:LINE {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:RLIMit:DISPlay:LINE?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the ripple limit line display (command/query).
<i>Target</i>	Active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Ripple limit line ON {OFF 0} Ripple limit line OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Limit Line

CALC:RLIM:FAIL?

```
CALCulate#[:SElected]:RLIMit:FAIL?
```

<i>Description</i>	Reads out the ripple limit test result (query only).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	1 Fail 0 Pass
<i>Equivalent Softkeys</i>	None

CALC:RLIM:REP?

CALCulate<Ch>[:SElected]:RLIMit:REPort[:DATA]?

<i>Description</i>	<p>Reads out the data array, which is the ripple limit test results (query only). The array size is $1+3N$, where N is the number of ripple limit bands. For the n-th point, where n from 1 to N:</p> <p><numeric 1> N total number of the bands</p> <p><numeric $3n-1$> n number of the band</p> <p><numeric $3n-0$> Ripple value in the n-th band</p> <p><numeric $3n+1$> Ripple limit test result in the n-th band:</p> <ul style="list-style-type: none"> • 0: Pass • 1: Fail
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric $3N+1$ > The data transfer format depends on the FORM:DATA command setting.
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	None

CALC:SMO

```
CALCulate<Ch>[:SElected]:SMOothing[:STATe] {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:SMOothing[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the trace smoothing function (command/query).
<i>Target</i>	Active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Trace smoothing ON {OFF 0} Trace smoothing OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Response > Smoothing

CALC:SMO:APER

```
CALCulate<Ch>[:SElected]:SMOothing:APERture <numeric>
```

```
CALCulate<Ch>[:SElected]:SMOothing:APERture?
```

<i>Description</i>	Sets or reads out the smoothing aperture, when performing smoothing function (command/query).
<i>Target</i>	Active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the smoothing aperture from 0.01 to 20
<i>Unit</i>	% (percent)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Response > Smo Aperture

CALC:TRAC:DATA:FDAT?

CALCulate<Ch>:TRACe<Tr>:DATA:FDATa?

<i>Description</i>	<p>Reads out the formatted data array (query only). The formatted data array is the data, whose processing is completed including the formatting as the last step. Such data represent the data trace values as they are shown on the screen. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> real number in rectangular format, real part in polar and Smith chart formats;</p> <p><numeric 2n> 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>The specified trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	CALC:DATA:FDAT
<i>Equivalent Softkeys</i>	None

CALC:TRAC:DATA:FMEM?

CALCulate<Ch>:TRACe<Tr>:DATA:FMEMemory?

<i>Description</i>	<p>Reads out the formatted memory array (query only). The formatted memory array is the data, whose processing is completed including the formatting as the last step. Such data represent the memory trace values as they are shown on the screen. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> real number in rectangular format, real part in polar and Smith chart formats;</p> <p><numeric 2n> 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>The specified trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	<p>If the memory is empty, an error occurs and the command is ignored.</p>
<i>Related Commands</i>	<p>CALC:DATA:FMEM</p>
<i>Equivalent Softkeys</i>	<p>None</p>

CALC:TRAC:DATA:SDAT?

CALCulate<Ch>:TRACe<Tr>:DATA:SDATa?

<i>Description</i>	<p>Reads out the corrected data array (query only). The corrected data array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> the real part of corrected measurement; <numeric 2n> the imaginary part of corrected measurement.</p>
<i>Target</i>	<p>The specified trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8} <Ch>={ [1] 2 3 4 5 6 7 8 9}</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	CALC:DATA:SDAT
<i>Equivalent Softkeys</i>	None

CALC:TRAC:DATA:SMEM?

CALCulate<Ch>:TRACe<Tr>:DATA:SMEMory?

<i>Description</i>	<p>Reads out the corrected memory array (query only). The corrected memory array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric 2n-1> the real part of corrected measurement memory; <numeric 2n> the imaginary part of corrected measurement memory.</p>
<i>Target</i>	<p>The specified trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric 2N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Notes</i>	<p>If the memory is empty, an error occurs and the command is ignored.</p>
<i>Related Commands</i>	<p>CALC:DATA:SMEM</p>
<i>Equivalent Softkeys</i>	<p>None</p>

CALC:TRAN:TIME

```
CALCulate<Ch>[:SElected]:TRANsform:TIME[:TYPE] <char>
```

```
CALCulate<Ch>[:SElected]:TRANsform:TIME[:TYPE]?
```

<i>Description</i>	Selects the transformation type for the time domain transformation function: bandpass response or direct current circuit (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char>specifies the transformation type: BPASs : Bandpass LPASs : Lowpass
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{BPAS LPAS }
<i>Preset Value</i>	BPAS
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Response Type > { Bandpass Lowpass Step Lowpass Impulse }

CALC:TRAN:TIME:CENT

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:CENTer <time>
```

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:CENTer?
```

<i>Description</i>	Sets or reads out the time domain center value, when the time domain transformation function is turned ON (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the time domain center value, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Center

CALC:TRAN:TIME:IMP:WIDT

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:IMPulse:WIDTh <time>
```

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:IMPulse:WIDTh?
```

<i>Description</i>	Sets or reads out the impulse width (time domain transformation resolution), coupled with the Kaiser–Bessel window shape β parameter. The impulse width setting changes the β parameter, and setting of β parameter changes the impulse width (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the impulse width, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Impulse Width (when the transformation type is set to Bandpass or Lowpass Impulse)

CALC:TRAN:TIME:KBES

CALCulate<Ch>[:SElected]:TRANSform:TIME:KBESsel <numeric>

CALCulate<Ch>[:SElected]:TRANSform:TIME:KBESsel?

<i>Description</i>	Sets or reads out the β parameter, which controls the Kaiser–Bessel window shape, when performing time domain transformation (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> β parameter from 0 to 13
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	6
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Kaiser Beta

CALC:TRAN:TIME:LPFR

CALCulate<Ch>[:SElected]:TRANSform:TIME:LPFRrequency

<i>Description</i>	Changes the frequency range to match with the lowpass type of the time domain transformation function (no query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Set Frequency Low Pass

CALC:TRAN:TIME:REFL:TYPE

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:REFlection:TYPE <char>
```

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:REFlection:TYPE?
```

<i>Description</i>	Selects the reflection distance either one way or round trip for the time domain transformation function. (command/query)
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> Choose from: RTRip : Round Trip OWAY : One Way
<i>Query Response</i>	{ RTR OWAY }
<i>Preset Value</i>	RTR
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Reflection Type > Round Trip One Way

CALC:TRAN:TIME:SPAN

CALCulate<Ch>[:SElected]:TRANsform:TIME:SPAN <time>

CALCulate<Ch>[:SElected]:TRANsform:TIME:SPAN?

<i>Description</i>	Sets or reads out the time domain span value, when the time domain transformation function is turned ON (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the time domain span value, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	2e-8
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Span

CALC:TRAN:TIME:STAR

CALCulate<Ch>[:SElected]:TRANsform:TIME:STARt <time>

CALCulate<Ch>[:SElected]:TRANsform:TIME:STARt?

<i>Description</i>	Sets or reads out the time domain start value, when the time domain transformation function is turned ON (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the time domain start value, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	-1e-8
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Start

CALC:TRAN:TIME:STAT

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:STATe {ON|OFF|1|0}
```

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the time domain transformation function (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Time domain transformation ON {OFF 0} Time domain transformation OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Time Domain

CALC:TRAN:TIME:STEP:RTIM

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:STEP:RTIME <time>
```

```
CALCulate<Ch>[:SElected]:TRANSform:TIME:STEP:RTIME?
```

<i>Description</i>	Sets or reads out the rise time of the step signal (time domain transformation resolution), coupled with the Kaiser-Bessel window shape β parameter. The impulse width setting changes the β parameter, and setting of β parameter changes the impulse width (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the impulse width, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Impulse Width (when the transformation type is set to Lowpass Step)

CALC:TRAN:TIME:STIM

```
CALCulate<Ch>[:SElected]:TRANsform:TIME:STIMulus <char>
```

```
CALCulate<Ch>[:SElected]:TRANsform:TIME:STIMulus?
```

<i>Description</i>	Selects the stimulus type for the time domain transformation function: impulse or step (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> specifies the stimulus type: IMPulse : Impulse STEP : Step
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ IMP STEP }
<i>Preset Value</i>	IMP
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Response Type > { Bandpass Lowpass Step Lowpass Impulse }

CALC:TRAN:TIME:STOP

CALCulate<Ch>[:SElected]:TRANsform:TIME:STOP <time>

CALCulate<Ch>[:SElected]:TRANsform:TIME:STOP?

<i>Description</i>	Sets or reads out the time domain stop value, when the time domain transformation function is turned ON (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the time domain stop value, the range varies depending on the specified frequency range and the number of points
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	+1e-8
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Stop

CALC:TRAN:TIME:UNIT

CALCulate<Ch> [:SElected]:TRANSform:TIME:UNIT <char>

CALCulate<Ch> [:SElected]:TRANSform:TIME:UNIT?

<i>Description</i>	Selects the the transformation unit for the time domain transformation function: seconds, meters, feet. (command/query)
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> choose from: SEC : Seconds MET : Meters FEET : Feet
<i>Query Response</i>	{ SEC MET FEET }
<i>Preset Value</i>	SEC
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Unit > Time, ns Metric, m Imperial, ft

DISP:COL:BACK

```
DISPlay:COLor:BACK <numeric 1>,<numeric 2>,<numeric 3>
```

```
DISPlay:COLor:BACK?
```

<i>Description</i>	Sets or reads out the background color for trace display (command/query).
<i>Parameter</i>	<p><numeric 1> Red value R from 0 to 255;</p> <p><numeric 2> Green value G from 0 to 255;</p> <p><numeric 3> Blue value B from 0 to 255.</p>
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Preset Value</i>	0, 0, 0
<i>Equivalent Softkeys</i>	Display > Color > Background

DISP:COL:GRAT

```
DISPlay:COLor:GRATicule <numeric 1>,<numeric 2>,<numeric 3>
```

```
DISPlay:COLor:GRATicule?
```

<i>Description</i>	Sets or reads out the grid and the graticule label color for trace display (command/query).
<i>Parameter</i>	<p><numeric 1> Red value R from 0 to 255;</p> <p><numeric 2> Green value G from 0 to 255;</p> <p><numeric 3> Blue value B from 0 to 255.</p>
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Preset Value</i>	160, 160, 164
<i>Equivalent Softkeys</i>	Display > Color > Grid

DISP:COL:RES

```
DISPlay:COLor:RESet
```

<i>Description</i>	Restores the display settings to the default values (no query).
<i>Equivalent Softkeys</i>	Display > Set Defaults

DISP:COL:TRAC:DATA

```
DISPlay:COLor:TRACe<Tr>:DATA <numeric 1>,<numeric 2>,<numeric 3>
```

```
DISPlay:COLor:TRACe<Tr>:DATA?
```

<i>Description</i>	Sets or reads out the data trace color (command/query).
<i>Target</i>	Trace <Tr>, <Tr>={ [1] 2 3 4 5 6 7 8 }
<i>Parameter</i>	<numeric 1> Red value R from 0 to 255; <numeric 2> Green value G from 0 to 255; <numeric 3> Blue value B from 0 to 255.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Preset Value</i>	Varies depending on the trace number.
<i>Equivalent Softkeys</i>	Display > Color > Data Trace

DISP:COL:TRAC:MEM

```
DISPlay:COLor:TRACe<Tr>:MEMory <numeric 1>,<numeric 2>,<numeric 3>
```

```
DISPlay:COLor:TRACe<Tr>:MEMory?
```

<i>Description</i>	Sets or reads out the memory trace color (command/query).
<i>Target</i>	Trace <Tr>, <Tr>={ [1] 2 3 4 5 6 7 8 }
<i>Parameter</i>	<numeric 1> Red value R from 0 to 255; <numeric 2> Green value G from 0 to 255; <numeric 3> Blue value B from 0 to 255.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Preset Value</i>	Varies depending on the trace number.
<i>Equivalent Softkeys</i>	Display > Color > Memory Trace

DISP:ENAB

```
DISPlay:ENABle {OFF|ON|0|1}
```

```
DISPlay:ENABle?
```

<i>Description</i>	Turns ON/OFF the display update (command/query).
<i>Parameter</i>	{ON 1} ON {OFF 0} OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Display > Update

DISP:FSIG

DISPlay:FSIGn {ON|OFF|1|0}

DISPlay:FSIGn?

<i>Description</i>	Sets or reads out the ON/OFF state of the Fail sign display, when performing limit test or ripple limit test (command/query).
<i>Parameter</i>	{ON 1} Fail sign display ON {OFF 0} Fail sign display OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Fail Sign Analysis > Ripple Limit > Fail Sign

DISP:IMAG

DISPlay:IMAGe <char>

DISPlay:IMAGe?

<i>Description</i>	Sets or reads out the inverted color display of the data traces (command/query).
<i>Parameter</i>	<char> choose from: NORMal : Normal display INVert : Inverted color display
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{NORM INV}
<i>Preset Value</i>	NORM
<i>Equivalent Softkeys</i>	Display > Invert Color

SYST:HIDE

SYSTem:HIDE

<i>Description</i>	Hides the instrument GUI (no query).
<i>Related Commands</i>	SYST:SHOW
<i>Equivalent Softkeys</i>	None

SYSTem:SHOW

DISPlay:SHOW

<i>Description</i>	Shows the instrument GUI hidden by the SYST:HIDE command (no query).
<i>Related Commands</i>	SYST:HIDE
<i>Equivalent Softkeys</i>	None

DISP:MAX

```
DISPlay:MAXimize {ON|OFF|1|0}
```

```
DISPlay:MAXimize?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the window maximization of the active channel (command/query).
<i>Parameter</i>	{ON 1} Maximization ON {OFF 0} Maximization OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Channel > Maximize Channel


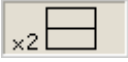

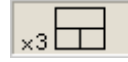
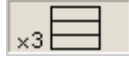

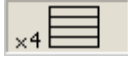
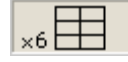
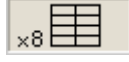
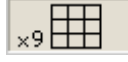
DISP:SPL

```
DISPlay:SPLit <numeric>
```

```
DISPlay:SPLit?
```

<i>Description</i>	Sets or reads out the layout of the channel windows on the screen (command/query). The channel window layout on the screen see below.
<i>Parameter</i>	<numeric> the number of the channel window layout from 1 to 10
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Channel > Allocate Channels

Channel window layout on the screen

1: 	2: 	3: 	4: 
5: 	6: 	7: 	8: 
9: 	10: 		

DISP:UPD

DISPlay:UPDate[:IMMediate]

<i>Description</i>	Updates the display once, when the display update is set to OFF by the DISP:ENAB command. (no query).
<i>Related Commands</i>	DISP:ENAB
<i>Equivalent Softkeys</i>	None

DISP:WIND:ACT

DISPlay:WINDow<Ch>:ACTivate

<i>Description</i>	Sets the active channel (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Notes</i>	At attempt to set to the active channel the channel, which is not displayed by the DISP:SPL command, an error occurs.
<i>Related Commands</i>	DISP:SPL
<i>Equivalent Softkeys</i>	Channel > Active Channel

DISP:WIND:ANN:MARK:ALIG

```
DISPlay:WINDow<Ch>:ANNotation:MARKer:ALIGn[:TYPE] <char>
```

```
DISPlay:WINDow<Ch>:ANNotation:MARKer:ALIGn[:TYPE]?
```

<i>Description</i>	Sets or reads out the alignment mode of the marker display position of each trace, when the only active trace display feature is turned OFF by the DISP:WIND:ANN:MARK:SING command (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> choose from: VERTical : Vertical alignment HORizontal : Horizontal alignment NONE : No alignment
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ NONE VERT HOR }
<i>Preset Value</i>	NONE
<i>Related Commands</i>	DISP:WIND:ANN:MARK:SING
<i>Equivalent Softkeys</i>	Markers > Properties > Align > { Vertical Horizontal OFF }

DISP:WIND:ANN:MARK:SING

```
DISPlay:WINDow<Ch>:ANNotation:MARKer:SINGle[:STATe]
{ON|OFF|1|0}
```

```
DISPlay:WINDow<Ch>:ANNotation:MARKer:SINGle[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the marker display for the active trace only (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Only active trace markers display ON {OFF 0} Only active trace markers display OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Markers > Properties > Active Only

DISP:WIND:MAX

```
DISPlay:WINDow<Ch>:MAXimize {ON|OFF|1|0}
```

```
DISPlay:WINDow<Ch>:MAXimize?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the active trace maximization of the specified channel (command/query).
<i>Parameter</i>	{ON 1} Maximization ON {OFF 0} Maximization OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Trace > Maximize Trace



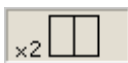

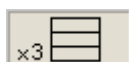

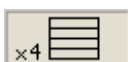
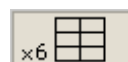

DISP:WIND:SPL

DISPlay:WINDow<Ch>:SPLit <numeric>

DISPlay:WINDow<Ch>:SPLit?

<i>Description</i>	Sets or reads out the layout of the graph in the channel window (command/query). The graph layout in the channel window see below.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the number of the graph layout from 1 to 9
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Trace > Allocate Traces

Graph layout in the channel window

1: 	2: 	3: 	4: 
5: 	6: 	7: 	8: 
9: 			

DISP:WIND:TITL

```
DISPlay:WINDow<Ch>:TITLe[:STATe] {ON|OFF|1|0}
```

```
DISPlay:WINDow<Ch>:TITLe[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the channel title display (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Channel title display ON {OFF 0} Channel title display OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Display > Title Label

DISP:WIND:TITL:DATA

```
DISPlay:WINDow<Ch>:TITLe:DATA <string>
```

```
DISPlay:WINDow<Ch>:TITLe:DATA?
```

<i>Description</i>	Sets or reads out the channel title label (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<string>, up to 256 characters (quoted string)
<i>Query Response</i>	<string>
<i>Preset Value</i>	""
<i>Equivalent Softkeys</i>	None

DISP:WIND:TRAC:ANN:MARK:MEM

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:MEMory <bool>
```

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:MEMory?
```

<i>Description</i>	Turns ON/OFF the state of the memory value display on the marker.
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<bool> Specifies the memory value display: { ON 1 } ON { OFF 0 } OFF
<i>Query Response</i>	{ 0 1 }
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Properties > Memory Value

DISP:WIND:TRAC:ANN:MARK:POS:X

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:X
<numeric>
```

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:X?
```

<i>Description</i>	Sets or reads out the display position of the marker value on the X-axis by a percentage of the display width (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8} <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Parameter</i>	<numeric> the display position of the marker value on the X-axis from 0 to 100
<i>Unit</i>	% (percent)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Properties > Data X Position

DISP:WIND:TRAC:ANN:MARK:POS:Y

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:Y
<numeric>
```

```
DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:Y?
```

<i>Description</i>	Sets or reads out the display position of the marker value on the Y-axis by a percentage of the display height (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8} <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Parameter</i>	<numeric> the display position of the marker value on the Y-axis from 0 to 100
<i>Unit</i>	% (percent)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Markers > Properties > Data Y Position

DISP:WIND:TRAC:Y:AUTO

```
DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:AUTO
```

<i>Description</i>	Executes the auto scale function for the trace (no query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8} <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Equivalent Softkeys</i>	Scale > Auto Scale

DISP:WIND:TRAC:Y:PDIV

```
DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:PDIVision <response>
```

```
DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:PDIVision?
```

<i>Description</i>	Sets or reads out the trace scale. Sets the scale per division, when the data format is the rectangular format. Sets the full scale value, when the data format is the Smith chart format or the polar format.
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<response> the scale value from 10E-18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	Varies depending on the format. Logarithmic Magnitude: 10 dB/Div Phase: 40 °/Div Expand Phase: 100 °/Div Group Delay: 10e-9 s/Div Smith Chart, Polar, SWR: 1 /Div Linear Magnitude: 0.1 /Div Real part, Imaginary part: 0.2 /Div
<i>Equivalent Softkeys</i>	Scale > Scale

DISP:WIND:TRAC:Y:RLEV

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:RLEVel <response>

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:RLEVel?

<i>Description</i>	Sets the value of the reference line (response value on the reference line). For the rectangular format only (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<response> the reference value from 10E-18 to 1E18
<i>Unit</i>	{ dB (decibel) ° (degree) s (second) }
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0 (except for SWR: 1)
<i>Equivalent Softkeys</i>	Scale > Ref Value

DISP:WIND:TRAC:Y:RPOS

```
DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RPOSition <numeric>
```

```
DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RPOSition?
```

<i>Description</i>	Sets the position of the reference line. For the rectangular format only (command/query).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the reference line position from 0 to the number of the scale divisions (set by the DISP:WIND:Y:DIV command, 10 by default)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	5 (except for SWR: 0)
<i>Equivalent Softkeys</i>	Scale > Ref Position

DISP:WIND:Y:DIV

```
DISPlay:WINDow<Ch>:Y[:SCALe]:DIVisions <numeric>
```

```
DISPlay:WINDow<Ch>:Y[:SCALe]:DIVisions?
```

<i>Description</i>	Sets the number of the vertical scale divisions. For the rectangular format only (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the number of the vertical scale divisions from 4 to 30
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	10
<i>Resolution</i>	2
<i>Equivalent Softkeys</i>	Scale > Divisions

FORM:BORD

FORMat:BORDer <char>

FORMat:BORDer?

<i>Description</i>	Sets or reads out the transfer order of each byte in data, when the binary data transfer format is set by the FORM:DATA command (command/query).
<i>Parameter</i>	<char> choose from: NORMal : Normal (from the most significant byte) SWAPped : Swapped (from the least significant byte)
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{NORM SWAP}
<i>Preset Value</i>	NORM
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	None

FORM:DATA

FORMat:DATA <char>

FORMat:DATA?

<i>Description</i>	<p>Sets or reads out the data transfer format, when responding to the following queries:</p> <p>CALC:DATA:FDAT? CALC:DATA:FMEM? CALC:DATA:SDAT? CALC:DATA:SMEM? CALC:FUNC:DATA? CALC:LIM:DATA? CALC:LIM:REP? CALC:LIM:REP:ALL? CALC:RLIM:DATA? CALC:RLIM:REP? SENS:FREQ:DATA? SENS:SEGM:DATA?</p> <p>(command/query)</p>
<i>Parameter</i>	<p><char> choose from:</p> <p>ASCIi : Character format</p> <p>REAL : Binary format (IEEE-64 floating point)</p> <p>REAL32 : Binary format (IEEE-32 floating point)</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{ASC REAL REAL32}
<i>Preset Value</i>	ASC
<i>Related Commands</i>	FORM:BORD
<i>Equivalent Softkeys</i>	None

HCOP

HCOPY[:IMMEDIATE]

<i>Description</i>	Prints out the image displayed on the screen without previewing (no query).
<i>Equivalent Softkeys</i>	System > Print > Print Embedded

HCOP:ABOR

HCOPY:ABORT

<i>Description</i>	Aborts the printout (no query).
<i>Equivalent Softkeys</i>	None

HCOP:DATE:STAM

HCOPY:DATE:STAMP {ON|OFF|1|0}

HCOPY:DATE:STAMP?

<i>Description</i>	Sets or reads out the ON/OFF state of the current date and time printout in the upper right corner (command/query).
<i>Parameter</i>	{ON 1} Date & time printout ON {OFF 0} Date & time printout OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	System > Print > Print Date & Time

HCOP:IMAG

HCOPy:IMAGe <char>

HCOPy:IMAGe?

<i>Description</i>	Sets or reads out the inverted color image printout (command/query).
<i>Parameter</i>	<char> choose from: NORMal : Normal printout INVert : Inverted color printout
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{NORM INV}
<i>Preset Value</i>	NORM
<i>Equivalent Softkeys</i>	System > Print > Invert Image

HCOP:PAIN

HCOPy:PAINT <char>

HCOPy:PAINT?

<i>Description</i>	Sets or reads out the color chart for the image printout (command/query).
<i>Parameter</i>	<char> choose from: COLor : Color printout GRAY : Grayscale printout BW : Black&white printout
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{COL GRAY BW}
<i>Preset Value</i>	BW
<i>Equivalent Softkeys</i>	System > Print > Print Color

INIT

INITiate<Ch>[:IMMediate]

<i>Description</i>	<p>Sets the channel to the single trigger mode. Before this command is sent, the channel must be in hold state, otherwise an error occurs and the command is ignored.</p> <p>On completion of the sweep, the channel goes back into the hold state.</p> <p>The command is completed before the end of the sweep. (no query)</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Notes</i>	<p>The sweep start in single trigger mode depends on the trigger source. If the trigger is set to internal, the sweep will start immediately upon the single mode activation. If the trigger is set otherwise, the sweep will start when the trigger signal is received.</p>
<i>Related Commands</i>	<p>TRIG:SOUR</p>
<i>Equivalent Softkeys</i>	<p>Stimulus > Trigger > Single</p>

INIT:CONT

```
INITiate<Ch>:CONTinuous {ON|OFF|1|0}
```

```
INITiate<Ch>:CONTinuous?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the continuous trigger initiation mode. If the continuous trigger initiation mode is set to OFF, the channel tuns to the hold state (command /query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Parameter</i>	{ON 1} Continuous trigger initiation mode ON {OFF 0} Continuous trigger initiation mode OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Notes</i>	The sweep start in continuous trigger initiation mode depends on the trigger source. If the trigger is set to internal, the sweeps will go immediately one after another. If the trigger is set otherwise, the sweep will start when the trigger signal is received.
<i>Related Commands</i>	TRIG:SOUR
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Continuous Stimulus > Trigger > Hold

MMEM:COPY

```
MMEMory:COPY <string1>,<string2>
```

<i>Description</i>	Copies a file (no query).
<i>Parameter</i>	<string1> Source file name (quoted string) <string2> Destination file name (quoted string)
<i>Equivalent Softkeys</i>	None

MMEM:DEL

MMEMory:DElete <string>

<i>Description</i>	Deletes a file (no query).
<i>Parameter</i>	<string> File name (quoted string)
<i>Equivalent Softkeys</i>	None

MMEM:LOAD

MMEMory:LOAD[:STATe] <string>

<i>Description</i>	Recalls the specified instrument state file. The file must be saved by the MMEM:STOR command (no query).
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \State subdirectory of the main directory will be searched for the file. The instrument state file has *.sta extension by default.
<i>Equivalent Softkeys</i>	System > Save/Recall > State

MMEM:LOAD:CHAN

MMEMory:LOAD:CHANnel[:STATe] <char>

<i>Description</i>	Recalls the instrument state for the active channel. The file must be saved in one of the four memory registers by the MMEM:STOR:CHAN command (no query).
<i>Target</i>	Active channel set by the DISP:WIND:ACT command
<i>Parameter</i>	<p><char> choose from:</p> <p>A : Recall from register A</p> <p>B : Recall from register B</p> <p>C : Recall from register C</p> <p>D : Recall from register D</p>
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	System > Save/Recall > Channel { State A B C D }

MMEM:LOAD:CKIT

MMEMory:LOAD:CKIT<Ck> <string>

<i>Description</i>	Recalls the definition file for the calibration kit. The file must be saved by the MMEM:STOR:CKIT command (no query).
<i>Target</i>	Calibration kit <Ck>, <Ck>={ [1] 2 3 4 5 6 7 8 9 10 11 }
<i>Parameter</i>	<string> Fine name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \CalKit subdirectory of the main directory will be searched for the file. The calibration kit definition file has *.ckd extension by default.
<i>Equivalent Softkeys</i>	None

MMEM:LOAD:LIM

MMEMory:LOAD:LIMit <string>

<i>Description</i>	Recalls the limit table file. The file must be saved by the MMEM:STOR:LIM command (no query).
<i>Target</i>	Active trace of the active channel, set by the CALC:PAR:SEL command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The limit table file has *.lim extension by default.
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line > Restore Limit Table

MMEM:LOAD:RLIM

MMEMory:LOAD:RLIMit <string>

<i>Description</i>	Recalls the ripple limit table file. The file must be saved by MMEM:STOR:RLIM the command (no query).
<i>Target</i>	Active trace of the active channel, set by the CALC:PAR:SEL command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The ripple limit file has *.rlm extension by default.
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit > Restore Ripple Limit Table

MMEM:LOAD:SEGM

MMEMory:LOAD:SEGMENT <string>

<i>Description</i>	Recalls the segment table file. The file must be saved by the MMEM:STOR:SEGM command (no query).
<i>Target</i>	Active channel, set by the DISP:WIND:ACT command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Segment subdirectory of the main directory will be searched for the file. The segment file has *.seg extension by default.
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Recall

MMEM:MDIR

MMEMory:MDIRectory <string>

<i>Description</i>	Creates a new directory (no query).
<i>Parameter</i>	<string> Directory full name (quoted string)
<i>Equivalent Softkeys</i>	None

MMEM:STOR

MMEMory:STORe[:STATe] <string>

<i>Description</i>	Saves the instrument state into a file (no query).
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \State subdirectory of the main directory will be searched for the file. The state file has *.sta extension by default.
<i>Equivalent Softkeys</i>	System > Save > State

MMEM:STOR:CHAN

MMEMory:STORe:CHANnel[:STATe] <char>

<i>Description</i>	Saves the instrument state of the items set for the active channel into one of the four memory registers (no query).
<i>Target</i>	Active channel set by the DISP:WIND:ACT command
<i>Parameter</i>	<p><char> choose from:</p> <p>A : Save to register A</p> <p>B : Save to register B</p> <p>C : Save to register C</p> <p>D : Save to register D</p>
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	System > Save/Recall > Channel > { State A B C D }

MMEM:STOR:CHAN:CLE

MMEMory:STORe:CHANnel:CLEar

<i>Description</i>	Clears the memory of the channel state saved by the MMEM:STOR:CHAN command (no query).
<i>Equivalent Softkeys</i>	System > Save/Recall > Channel > Clear States

MMEM:STOR:CKIT

MMEMory:STORe:CKIT<Ck> <string>

<i>Description</i>	Saves the definition file for the calibration kit (no query).
<i>Target</i>	Calibration kit <Ck>, <Ck>={ [1] 2 3 4 5 6 7 8 9 10 11 }
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \CalKit subdirectory of the main directory will be searched for the file. The calibration kit definition file has *.ckd extension by default.
<i>Equivalent Softkeys</i>	None

MMEM:STOR:FDAT

MMEMory:STORe:FDATa <string>

<i>Description</i>	Saves the CSV formatted data into a file (no query).
<i>Target</i>	Active trace of the active channel, set by the CALC:PAR:SEL command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \CSV subdirectory of the main directory will be searched for the file. The file has *.csv extension by default.
<i>Equivalent Softkeys</i>	System > Save > Save Trace Data

MMEM:STOR:IMAG

MMEMory:STORe:IMAGe <string>

<i>Description</i>	Saves the display image in BMP or PNG format into a file (no query).
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Image subdirectory of the main directory will be searched for the file. If the file has *.png extension, the file had PNG format, in all the other cases the file has BMP format.
<i>Equivalent Softkeys</i>	System > Print > Print Windows > Save as...

MMEM:STOR:LIM

MMEMory:STORe:LIMit <string>

<i>Description</i>	Saves the limit table into a file (no query).
<i>Target</i>	Active trace of the active channel, set by the CALC:PAR:SEL command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The file has *.lim extension by default.
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line > Save Limit Table

MMEM:STOR:RLIM

MMEMory:STORe:RLIMit <string>

<i>Description</i>	Saves the ripple limit table into a file (no query).
<i>Target</i>	Active trace of the active channel, set by the CALC:PAR:SEL command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The ripple limit file has *.rlm extension by default.
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit > Save Ripple Limit Table

MMEM:STOR:SEGM

MMEMory:STORe:SEGMENT <string>

<i>Description</i>	Save the segment table in a file (no query).
<i>Target</i>	Active channel, set by the DISP:WIND:ACT command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \Segment subdirectory of the main directory will be searched for the file. The segment file has *.seg extension by default.
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Save...

MMEM:STOR:SNP

MMEMory:STORe:SNP[:DATA] <string>

<i>Description</i>	<p>Saves the measured S-parameters of the active channel into a Touchstone file. The file type (1-port or 2-port) is set by the MMEM:STOR:SNP:TYPE:S1P and MMEM:STOR:SNP:TYPE:S2P commands. 1-port type file saves one reflection parameter: S11 or S22. 2-port type file saves all the four parameters: S11, S21, S12, S22.</p> <p>(no query)</p>
<i>Target</i>	Active channel, set by the DISP:WIND:ACT command
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \FixtureSim subdirectory of the main directory will be searched for the file. The 1-port measurement file has *.slp extension; the 2-port measurement file has *.s2p extension.
<i>Equivalent Softkeys</i>	System > Save > Touchstone File > Save

MMEM:STOR:SNP:FORM

```
MMEMory:STORe:SNP:FORMat <char>
```

```
MMEMory:STORe:SNP:FORMat?
```

<i>Description</i>	Sets the data format for the S-parameter saving by the MMEM:STOR:SNP command (command/query).
<i>Target</i>	Active channel, set by the DISP:WIND:ACT command
<i>Parameter</i>	<p><char> choose from:</p> <p>MA : Logarithmic Magnitude / Angle format</p> <p>DB : Linear Magnitude / Angle format</p> <p>RI : Real part /Imaginary part format</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{RI DB MA}
<i>Preset Value</i>	RI
<i>Equivalent Softkeys</i>	System > Save > Touchstone File > Format

MMEM:STOR:SNP:TYPE:S1P

```
MMEMory:STORe:SNP:TYPE:S1P <port>
```

```
MMEMory:STORe:SNP:TYPE:S1P?
```

<i>Description</i>	Sets the 1-port Touchstone file type (*.s1p) and the port number, when saving S-parameters by the MMEM:STOR:SNP command. (command/query)
<i>Parameter</i>	<port> port number from 1 to 2
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	System > Save > Touchstone File > Type > 1-Port (s1p)

MMEM:STOR:SNP:TYPE:S2P

```
MMEMory:STORe:SNP:TYPE:S2P <rcvport>, <srcport>
```

```
MMEMory:STORe:SNP:TYPE:S2P?
```

<i>Description</i>	Sets the 2-port Touchstone file type (*.s2p) and the port numbers, when saving S-parameters by the MMEM:STOR:SNP command. (command/query)
<i>Parameter</i>	<rcvport> the number of the receiver port from 1 to 2 <srcport> the number of the source port 1
<i>Query Response</i>	<numeric1>, <numeric2>
<i>Equivalent Softkeys</i>	System > Save > Touchstone File > Type > 2-Port (s2p)

MMEM:STOR:STYP

MMEMory:STORe:STYPe <char>

MMEMory:STORe:STYPe?

<i>Description</i>	Selects the type of the instrument or channel state saving by the MMEM:STOR:CHAN command (command/query).
<i>Parameter</i>	<p><char> choose from:</p> <p>STATE : Measurement conditions</p> <p>CState : Measurement conditions and calibration tables</p> <p>DState : Measurement conditions and data traces</p> <p>CDState : Measurement conditions, calibration tables and data traces</p>
<i>Out of Range</i>	An error occurs.
<i>Query Response</i>	{STAT CST DST CDST}
<i>Preset Value</i>	CST
<i>Equivalent Softkeys</i>	System > Save > Save Type

MMEM:TRAN?

MMEMory:TRANsfer? <string>

<i>Description</i>	Transfers the contents of a specified file from the instrument to the external computer (command/query).
<i>Parameter</i>	<string> the file name with the full path (quoted string)
<i>Query Response</i>	Block data transfer format. For example: #6001000 <binary block with length of 1000 bytes> #6 Symbol # introduces the data block. The next number indicates how many of the following digits describe the length of the data block; 001000 Length of the data block;
<i>Notes</i>	The file must be 20 Mbytes or less.
<i>Equivalent Softkeys</i>	None

OUTP

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

OUTPut[:STATe]?

<i>Description</i>	Sets or reads out the ON/OFF state of the stimulus signal output. Measurements cannot be performed when the stimulus signal output is set to OFF (command/query).
<i>Parameter</i>	{ON 1} Stimulus signal output ON {OFF 0} Stimulus signal output OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Stimulus > Power > RF Out Off

SENS:AVER

```
SENSe<Ch>:AVERAge[:STATe] {ON|OFF|1|0}
```

```
SENSe<Ch>:AVERAge[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the averaging function (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Parameter</i>	{ON 1} Averaging ON {OFF 0} Averaging OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Response > Averaging

SENS:AVER:CLE

```
SENSe<Ch>:AVERAge:CLEAr
```

<i>Description</i>	Restarts the averaging process, when averaging function is set to ON (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Related Commands</i>	SENS:AVER
<i>Equivalent Softkeys</i>	None

SENS:AVER:COUN

```
SENSe<Ch>:AVERage:COUNT <numeric>
```

```
SENSe<Ch>:AVERage:COUNT?
```

<i>Description</i>	Sets or reads out the averaging factor, when the averaging function is set to ON (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the averaging factor from 1 to 999
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	10
<i>Related Commands</i>	SENS:AVER
<i>Equivalent Softkeys</i>	Response > Avg Factor

SENS:BAND

```
SENSe<Ch>:BANDwidth[:RESolution] <frequency>
```

```
SENSe<Ch>:BANDwidth[:RESolution]?
```

SENS:BWID

```
SENSe<Ch>:BWIDth[:RESolution] <frequency>
```

```
SENSe<Ch>:BWIDth[:RESolution]?
```

<i>Description</i>	Sets or reads out the IF bandwidth (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the IF bandwidth value from 1 to 30000
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	10000
<i>Resolution</i>	In steps of 10, 30, 100, 300, 1000, 3000, 10000, 30000
<i>Equivalent Softkeys</i>	Response > IF Bandwidth

SENS:CORR:CLE

SENSe<Ch>:CORRection:CLEar

<i>Description</i>	Clears the calibration coefficient table (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF

```
SENSe<Ch>:CORRection:COEFFicient[:DATA]? <char>,
<rcvport>,<srcport>,<numeric list>
```

```
SENSe<Ch>:CORRection:COEFFicient[:DATA]? <char>,
<rcvport>,<srcport>
```

<i>Description</i>	<p>Writes or reads out the calibration coefficient data array (command/query). The array size is $2N$, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric $2n-1$> real part of the calibration coefficients</p> <p><numeric $2n$> imaginary part of the calibration coefficients</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Parameter</i>	<p><char> specifies the error term selection:</p> <p>ER : Reflection tracking</p> <p>ED : Directivity</p> <p>ES : Source match</p> <p>ET : Transmission tracking</p> <p>EX : Isolation</p> <p>EL : Load match</p> <p><rcvport> , the number of the receiver port from 1 to 2</p> <p><srcport> the number of the source port = 1</p> <p><numeric list> the calibration coefficient array</p> <p>When ES, ER, or ED is used, the numbers of the ports <rcvport> and <srcport> must be the same. When EL, ET, or EX is used, the numbers of the ports <rcvport> and <srcport> must be different.</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric 2N>
<i>Notes</i>	The written calibration coefficients become effective only after the SENS:CORR:COEF:SAVE command is executed.
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:METH:ERES

SENSe<Ch>:CORRection:COEFFicient:METHod:ERESponse
<rcvport>, <srcport>

<i>Description</i>	Selects the ports and sets the 1–path 2–port calibration type, when the written calibration coefficients are made effective by the SENS:CORR:COEF:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs. If the same port numbers are specified, an error occurs.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:METH:OPEN

```
SENSe<Ch>:CORRection:COEFFicient:METhod[:RESPonse]:OPEN <port>
```

<i>Description</i>	Selects the port and sets the response calibration (Open) type, when the written calibration coefficients are made effective by the SENS:CORR:COEF:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	An error occurs..
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:METH:SHOR

```
SENSe<Ch>:CORRection:COEFFicient:METhod[:RESPonse]:SHORt
<port>
```

<i>Description</i>	Selects the port and sets the response calibration (Short) type, when the written calibration coefficients are made effective by the SENS:CORR:COEF:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	An error occurs.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:METH:SOLT1

SENSe<Ch>:CORRection:COEFFicient:METhod:SOLT1 <port>

<i>Description</i>	Selects the port and sets the full 1–port calibration type, when the written calibration coefficients are made effective by the SENS:CORR:COEF:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	An error occurs.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:METH:THRU

```
SENSe<Ch>:CORRection:COEFFicient:METHod[:RESPonse]:THRU
<rcvport>, <srcport>
```

<i>Description</i>	Selects the ports and sets the response calibration (Thru) type, when the written calibration coefficients are made effective by the SENS:CORR:COEF:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs. If the same port numbers are specified, an error occurs.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COEF:SAVE

SENSe<Ch>:CORRection:COEFFicient:SAVE

<i>Description</i>	<p>Enables the written calibration coefficients depending on the selected calibration type. On completion of the command the error correction automatically turns ON.</p> <p>At the attempt to execute this command before all the needed calibration coefficients are written, an error occurs and the command is ignored (no query).</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Related Commands</i>	<p>Calibration type selection: SENS:CORR:COEF:METH:ERES SENS:CORR:COEF:METH:OPEN SENS:CORR:COEF:METH:SHOR SENS:CORR:COEF:METH:THRU SENS:CORR:COEF:METH:SOLT1</p> <p>Calibration coefficient writing: SENS:CORR:COEF</p>
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:CKIT

```
SENSe<Ch>:CORRection:COLLect:CKIT[:SElect] <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT[:SElect]?
```

<i>Description</i>	Sets or reads out the number of the selected calibration kit (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the number of the calibration kit from 1 to 11
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Cal Kit n

SENS:CORR:COLL:CKIT:LAB

```
SENSe<Ch>:CORRection:COLLect:CKIT:LABel <string>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:LABel?
```

<i>Description</i>	Sets or reads out the calibration kit label (command/query).
<i>Target</i>	Calibration kit, selected for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<string>, up to 254 characters (quoted string)
<i>Query Response</i>	<string>
<i>Preset Value</i>	Varies depending on the number of the calibration kit.
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Label

SENS:CORR:COLL:CKIT:RES

SENSe<Ch>:CORRection:COLLect:CKIT:RESet

<i>Description</i>	Resets the calibration kit to the factory settings (no query).
<i>Target</i>	Calibration kit, selected for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	Calibration > Restore Cal Kit

SENS:CORR:COLL:CKIT:STAN:C0

SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C0 <numeric>

SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C0?

<i>Description</i>	Sets or reads out the C0 value for the open calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the C0 value from -1E18 to 1E18
<i>Unit</i>	1E-15 F (Farad)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Capacitance > C0 10⁻¹⁵ F

SENS:CORR:COLL:CKIT:STAN:C1

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C1 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C1?
```

<i>Description</i>	Sets or reads out the C1 value for the open calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the C1 value from -1E18 to 1E18
<i>Unit</i>	1E-27 F/Hz (Farad/Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Capacitance > C1 10⁻²⁷ F/Hz

SENS:CORR:COLL:CKIT:STAN:C2

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C2 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C2?
```

<i>Description</i>	Sets or reads out the C2 value for the open calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 ... N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the C2 value from -1E18 to 1E18
<i>Unit</i>	1E-36 F/Hz ² (Farad/Hertz ²)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Capacitance > C2 10⁻³⁶ F/Hz²

SENS:CORR:COLL:CKIT:STAN:C3

SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C3 <numeric>

SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:C3?

<i>Description</i>	Sets or reads out the C3 value for the open calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 ... N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the C3 value from -1E18 to 1E18
<i>Unit</i>	1E-45 F/Hz ³ (Farad/Hertz ³)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Capacitance > C3 10⁻⁴⁵ F/Hz³

SENS:CORR:COLL:CKIT:STAN:DEL

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:DELay <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:Delay?
```

<i>Description</i>	Sets or reads out the offset delay value for the calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the offset delay value form -1E18 to 1E18
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Offset Delay

SENS:CORR:COLL:CKIT:STAN:L0

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L0 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L0?
```

<i>Description</i>	Sets or reads out the L0 value for the short calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 ... N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the L0 value from -1E18 to 1E18
<i>Unit</i>	1E-12 H (Henry)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Inductance > L0 10⁻¹² H

SENS:CORR:COLL:CKIT:STAN:L1

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L1 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L1?
```

<i>Description</i>	Sets or reads out the L1 value for the short calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 ... N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the L1 value from -1E18 to 1E18
<i>Unit</i>	1E-24 H/Hz (Henry/Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Inductance > L1 10⁻²⁴ H/Hz

SENS:CORR:COLL:CKIT:STAN:L2

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L2 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L2?
```

<i>Description</i>	Sets or reads out the L2 value for the short calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the L2 value from -1E18 to 1E18
<i>Unit</i>	1E-33 H/Hz ² (Henry/Hertz ²)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Inductance > L2 10⁻³³ H/Hz²

SENS:CORR:COLL:CKIT:STAN:L3

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L3 <numeric>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:L3?
```

<i>Description</i>	Sets or reads out the L3 value for the short calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 ... N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<numeric> the L3 value from -1E18 to 1E18
<i>Unit</i>	1E-42 H/Hz ³ (Henry/Hertz ³)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Inductance > L3 10⁻⁴² H/Hz³

SENS:CORR:COLL:CKIT:STAN:LAB

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:LABel <string>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:LABel?
```

<i>Description</i>	Sets or reads out the label for the calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<string>, up to 254 characters (quoted string)
<i>Query Response</i>	<string>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Label

SENS:CORR:COLL:CKIT:STAN:LOSS

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:LOSS <loss>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:LOSS?
```

<i>Description</i>	Sets or reads out the offset loss value for the calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<loss> the offset loss value from -1E18 to 1E18
<i>Unit</i>	Ω/s (Ohm/second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Offset Loss

SENS:CORR:COLL:CKIT:STAN:TYPE

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:TYPE <char>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:TYPE?
```

<i>Description</i>	Sets or reads out the type of calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<char> choose from: OPEN : Open SHORT : Short LOAD : Load THRU : Thru
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	{ OPEN SHOR LOAD THRU }
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > STD Type

SENS:CORR:COLL:CKIT:STAN:Z0

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:Z0 <impedance>
```

```
SENSe<Ch>:CORRection:COLLect:CKIT:STAN<Std>:Z0?
```

<i>Description</i>	Sets or reads out the offset Z0 value for the calibration standard (command/query).
<i>Target</i>	Standard <Std> of the calibration kit specified for channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Std>={ [1] 2 . . . N }, where N – the number of the standards in the calibration kit
<i>Parameter</i>	<impedance> the offset Z0 value from –1E18 to 1E18
<i>Unit</i>	Ω (Ohm)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	50 or 75, depending on the selected calibration kit
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Offset Z0

SENS:CORR:COLL:CLEAR

```
SENSe<Ch>:CORRection:COLLect:CLear
```

<i>Description</i>	Clears the measurement values of the calibration standards (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	Calibration > Calibrate > { Response (Open) Response (Short) Response (Thru) One Path 2–Port Cal Full 1–Port Cal } > Cancel

SENS:CORR:COLL:ECAL:CHECK:EXEC

```
SENSe<Ch>:CORRection:COLLect:ECAL:CHECK:EXECute <numeric1>,
<numeric2>
```

<i>Description</i>	Executing confidence check
<i>Target</i>	AutoCal module (Ch – arbitrary number 1 to 9)
<i>Equivalent Softkeys</i>	Calibration > Autocalibration > Confidence Check

SENS:CORR:COLL:ECAL:ORI:EXEC

```
SENSe<Ch>:CORRection:COLLect:ECAL:ORIEntation:EXECute
<numeric1>, <numeric2>
```

<i>Description</i>	Executing AutoCal orientation to the ports Rcv, Scr
<i>Target</i>	AutoCal module
<i>Equivalent Softkeys</i>	Calibration > Autocalibration > Perform Auto-Orientation

SENS:CORR:COLL:ECAL:ORI:STAT

```
SENSe:CORRection:COLLect:ECAL:ORiEntation:STATe {ON|OFF|1|0}
```

```
SENSe:CORRection:COLLect:ECAL:ORiEntation:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the Auto-Orientation function used when executing AutoCal (command/query).
<i>Target</i>	AutoCal
<i>Parameter</i>	{ON 1} Auto-Orientation function ON {OFF 0} Auto-Orientation function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Auto-Orientation

SENS:CORR:COLL:ECAL:PATH

```
SENSe:CORRection:COLLect:ECAL:PATH <numeric1>,<numeric2>
```

```
SENSe:CORRection:COLLect:ECAL:PATH? <numeric1>
```

<i>Description</i>	Sets or reads out the AutoCal module port number which is connected to a specified port of Network Analyzer (command/query).
<i>Target</i>	AutoCal
<i>Parameter</i>	<p><numeric1> Network Analyzer Port Number, <numeric2> AutoCal Module Port Number,</p> <p>1 : Port A of AutoCal Module 2 : Port B of AutoCal Module 3 : Port C of AutoCal Module 4 : Port D of AutoCal Module</p>
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Port 1 – Port A Port 1 – Port B Port 1 – Port C Port 1 – Port D

SENS:CORR:COLL:ECAL:SOLT1

SENSE<Ch>:CORREction:COLLect:ECAL:SOLT1 <port>

<i>Description</i>	Executes Full 1-port calibration of the specified port of specified channel (Ch) using the AutoCal module (command only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> : Port Number 1
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Full 1-Port Calibration

SENS:CORR:COLL:ECAL:SOLT2

SENSE<Ch>:CORREction:COLLect:ECAL:SOLT1 <port2>,<port1>

<i>Description</i>	Executes One Path 2-Port calibration of the specified port of specified channel (Ch) using the AutoCal module (command only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port1> the number of the receiver port (always 1) <port2> the number of the source port (always 2)
<i>Equivalent Softkeys</i>	Calibration > AutoCal > One Path 2-Port Calibration

SENS:CORR:COLL:ECAL:UCH

SENSe:CORRection:COLLect:ECAL:UChar <char>

SENSe:CORRection:COLLect:ECAL:UChar?

<i>Description</i>	Sets or reads out the Characteristic used when executing AutoCal (factory or user characterization). (command/query).
<i>Target</i>	AutoCal
<i>Parameter</i>	<p><char> specifies the stimulus type:</p> <p>CHAR0 : factory characterization</p> <p>CHAR1 : user characterization 1</p> <p>CHAR2 : user characterization 2</p> <p>CHAR3 : user characterization 3</p>
<i>Query Response</i>	{ CHAR0 CHAR1 CHAR2 CHAR3 }
<i>Preset Value</i>	CHAR0
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Characterization

SENS:CORR:COLL:LOAD

SENSe<Ch>:CORRection:COLLect[:ACQuire]:LOAD <port>

<i>Description</i>	Measures the calibration data of the load standard for the specified port (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Notes</i>	The command start the measurement for the channel independently of the trigger and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Load (Optional) Calibration > Calibrate > Response (Short) > Load (Optional) Calibration > Calibrate > Full 1-Port Cal > Load Calibration > Calibrate > One Path 2-Port Cal > Load

SENS:CORR:COLL:METH:ERES

```
SENSe<Ch>:CORRection:COLLect:METhod:ERESponse <rcvport>,
<srcport>
```

<i>Description</i>	Selects the ports and sets the one path 2-port calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs. If the same port numbers are specified, an error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:METH:OPEN

```
SENSe<Ch>:CORRection:COLLect:MEthod[:RESPonse]:OPEN <port>
```

<i>Description</i>	Selects the port and sets the response calibration (Open) type for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:METH: SHOR

```
SENSe<Ch>:CORRection:COLLect:MEthod[:RESPonse]:SHORT <port>
```

<i>Description</i>	Selects the port and sets the response calibration (Short) type for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:METH:SOLT1

SENSe<Ch>:CORRection:COLLect:METhod:SOLT1 <port>

<i>Description</i>	Selects the port and sets the full 1–port calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:METH:THRU

```
SENSe<Ch>:CORRection:COLLect:METhod[:RESPOse]:THRU <rcvport>,
<srcport>
```

<i>Description</i>	Selects the ports and sets the response calibration (Thru) type for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs. If the same port numbers are specified, an error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:METH:TYPE?

SENSe<Ch>:CORREction:COLLect:METhod:TYPE?

<i>Description</i>	Reads out the calibration type selected for the calculation of the calibration coefficients on completion of the calibration executed by the SENS:CORR:COLL:SAVE command (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	RESPO" : Response (Open) RESPS : Response (Short) RESPT : Response (Thru) SOLT1 : Full 1-port calibration 1PATH : One path 2-port calibration NONE : Not defined
<i>Equivalent Softkeys</i>	None

SENS:CORR:COLL:OPEN

SENSe<Ch>:CORRection:COLLect[:ACQuire]:OPEN <port>

<i>Description</i>	Measures the calibration data of the open standard for the specified port (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Notes</i>	The command start the measurement for the channel independently of the trigger and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Open Calibration > Calibrate > Full 1-Port Cal > Open Calibration > Calibrate > One Path 2-Port Cal > Open

SENS:CORR:COLL:SAVE

SENSe<Ch>:CORRection:COLLect:SAVE

<i>Description</i>	<p>Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type. On completion of the command, all the calibration standards measurements are cleared and the error correction automatically turns ON.</p> <p>Before executing this command it is necessary to select calibration type by one of commands SENS:CORR:COLL:METH:XXXX.</p> <p>At the attempt to execute this command before all the needed standards are measured, an error occurs and the command is ignored (no query).</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Related Commands</i>	<p>Calibration type selection: SENS:CORR:COLL:METH:OPEN SENS:CORR:COLL:METH:SHOR SENS:CORR:COLL:METH:THRU SENS:CORR:COLL:METH:ERES SENS:CORR:COLL:METH:SOLT1</p> <p>Calibration standards measurement: SENS:CORR:COLL:LOAD SENS:CORR:COLL:OPEN SENS:CORR:COLL:SHOR SENS:CORR:COLL:THRU</p>
<i>Equivalent Softkeys</i>	<p>Calibration > Calibrate > { Response (Open) Response (Short) Response (Thru) One Path 2-Port Cal Full 1-Port Cal } > Apply</p>

SENS:CORR:COLL:SHOR

SENSe<Ch>:CORRection:COLLect[:ACQuire]:SHORT <port>

<i>Description</i>	Measures the calibration data of the short standard for the specified port (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<port> the number of the port from 1 to 2
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Notes</i>	The command start the measurement for the channel independently of the trigger and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Short) > Short Calibration > Calibrate > Full 1-Port Cal > Short Calibration > Calibrate > One Path 2-Port Cal > Short

SENS:CORR:COLL:THRU

```
SENSe<Ch>:CORRection:COLLect[:ACQuire]:THRU <rcvport>,
<srcport>
```

<i>Description</i>	Measures the calibration data of the thru standard between the receiver port <rcvport> and the source port <srcport> (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs. If the same port numbers are specified, an error occurs. The command is ignored.
<i>Notes</i>	The command start the measurement for the channel independently of the trigger and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Thru) > Thru Calibration > Calibrate > One Path 2-Port Cal > Thru

SENS:CORR:EXT

```
SENSe<Ch>:CORRection:EXTension[:STATe] {ON|OFF|1|0}
```

```
SENSe<Ch>:CORRection:EXTension[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the port extension function (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Port extension function ON {OFF 0} Port extension function OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Extension

SENS:CORR:EXT:PORT:FREQ

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:FREQuency{ [1] | 2 }
<frequency>
```

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:FREQuency{ [1] | 2 }?
```

<i>Description</i>	Sets or reads out the values of the frequency 1 and frequency 2 to calculate the loss for the port extension function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<frequency> the frequency value from 3E5 to 1.3E9 (Planar – TR1300/1), from 2E4 to 4.8E9 (TR5048), from 2E4 to 3.0E9 (TR7530)
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1E9
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > { Freq1 Freq2 }

SENS:CORR:EXT:PORT:INCL

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:INCLude{ [1] | 2 } [ :STATe]
{ON|OFF|1|0}
```

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:INCLude{ [1] | 2 }
[ :STATe] ?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the compensation of the loss 1 and loss 2 for the port extension function (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Loss compensation ON {OFF 0} Loss compensation OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > { Loss1 Loss2 }

SENS:CORR:EXT:PORT:LDC

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LDC <loss>
```

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LDC?
```

<i>Description</i>	Sets or reads out the loss value at DC for the port extension function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<loss> the loss value from -200 to 200
<i>Unit</i>	dB (decibel)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > Loss at DC

SENS:CORR:EXT:PORT:LOSS

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LOSS{ [1] | 2 } <loss>
```

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LOSS{ [1] | 2 }?
```

<i>Description</i>	Sets or reads out the values of the loss 1 and loss 2 for the port extension function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<loss> the loss value from -200 to 200
<i>Unit</i>	dB (decibel)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > { Loss1 Loss2 }

SENS:CORR:EXT:PORT:TIME

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:TIME <time>
```

```
SENSe<Ch>:CORRection:EXTension:PORT<Pt>:TIME?
```

<i>Description</i>	Sets or reads out the electrical delay value for the port extension function (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<time> the electrical delay value from -10 to 10
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > { Extension Port1 Extension Port2 }

SENS:CORR:IMP

```
SENSe:CORRection:IMPedance[:INPut][:MAGNitude] <impedance>
```

```
SENSe:CORRection:IMPedance[:INPut][:MAGNitude]?
```

<i>Description</i>	Sets or reads out the system impedance Z0 (command/query)
<i>Parameter</i>	<impedance> the Z0 value from 0.001 to 1000
<i>Unit</i>	Ω (Ohm)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	50
<i>Equivalent Softkeys</i>	Calibration > System Z0

SENS:CORR:STAT

```
SENSe<Ch>:CORRection:STATe {ON|OFF|1|0}
```

```
SENSe<Ch>:CORRection:STATe?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the error correction (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} Error correction ON {OFF 0} Error correction OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Calibration > Correction

SENS:CORR:TRAN:TIME:FREQ

```
SENSe<Ch>:CORRection:TRANsform:TIME:FREQuency <frequency>
```

```
SENSe<Ch>:CORRection:TRANsform:TIME:FREQuency?
```

<i>Description</i>	Sets or reads out the frequency value at which the cable loss specified for the cable correction function, when the time domain transformation function is turned ON. (command/query)
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the frequency value.
<i>Unit</i>	Hz (Hertz)
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1 GHz
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Frequency

SENS:CORR:TRAN:TIME:LOSS

```
SENSe<Ch>:CORRection:TRANsform:TIME:LOSS <numeric>
```

```
SENSe<Ch>:CORRection:TRANsform:TIME:LOSS?
```

<i>Description</i>	Sets or reads out the cable loss value for the cable correction function, when the time domain transformation function is turned ON. (command/query)
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the cable loss value.
<i>Unit</i>	dB/m (decibell / meter)
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0 dB/m
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Cable Loss

SENS:CORR:TRAN:TIME:RVEL

```
SENSe<Ch>:CORRection:TRANsform:TIME:RVELocity <numeric>
```

```
SENSe<Ch>:CORRection:TRANsform:TIME:RVELocity?
```

<i>Description</i>	Sets or reads out the cable relative wave speed velocity for the cable correction function, when the time domain transformation function is turned ON. (command/query)
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the cable velocity factor.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1.0
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Velocity Factor

SENS:CORR:TRAN:TIME:STAT

```
SENSe<Ch>:CORRection:TRANsform:TIME:STATe <bool>
```

```
SENSe<Ch>:CORRection:TRANsform:TIME:STATe?
```

<i>Description</i>	Turns ON/OFF the cable correction when the time domain transformation function is turned ON. (command/query)
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	Specifies state of the cable correction: {ON 1} : ON {OFF 0} : OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Cable Correction

SENS:CORR:TYPE?

SENSe<Ch>:CORRection:TYPE<Tr>?

<i>Description</i>	Reads out the applied calibration type and the port numbers for the specified trace (query only).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1] 2 3 4 5 6 7 8 } <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	{ RESPO RESPS RESPT SOLT1 SOLT2 1PATH NONE }, <srcport>, <rcvport> Where: RESPO : Response (Open) RESPS : Response (Short) RESPT : Response (Thru) SOLT1 : Full 1–port calibration 1PATH : One path 2–port calibration NONE : Not defined <rcvport>, the number of the receiver port from 1 to 2 <srcport>, the number of the source port = 1
<i>Equivalent Softkeys</i>	None

SENS:FREQ

```

SENSe<Ch>:FREQuency[:CW] <frequency>
SENSe<Ch>:FREQuency[:FIXed] <frequency>
SENSe<Ch>:FREQuency[:CW]?
SENSe<Ch>:FREQuency[:FIXed]?

```

<i>Description</i>	Sets or reads out the fixed frequency for the power sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the CW frequency value from 3E5 to 1.3E9 (Planar – TR1300/1), from 2E4 to 4.8E9 (TR5048), from 2E4 to 3.0E9 (TR7530).
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	6.5015E8 (Planar – TR1300/1), 2.40001E9 (TR5048), 1.50001E9 (TR7530).
<i>Equivalent Softkeys</i>	Stimulus > Power > CW Freq

SENS:FREQ:DATA?

SENSe<Ch>:FREQuency:DATA?

<i>Description</i>	<p>Reads out the array of the measurement points frequency (query only).</p> <p>The array size is N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric n> the frequency value at the n-th measurement point</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	None

SENS:FREQ:CENT

```
SENSe<Ch>:FREQuency:CENTer <frequency>
```

```
SENSe<Ch>:FREQuency:CENTer?
```

<i>Description</i>	Sets or reads out the stimulus center value of the sweep range for linear or logarithmic sweep type (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the stimulus center value from 3E5 to 1.3E9 (Planar – TR1300/1), from 2E4 to 4.8E9 (TR5048), from 2E4 to 3.0E9 (TR7530).
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	6.5015E8 (Planar - TR1300/1), 2.40001E9 (TR5048), 1.50001E9 (TR7530)
<i>Equivalent Softkeys</i>	Stimulus > Center

SENS:FREQ:SPAN

```
SENSe<Ch>:FREQuency:SPAN <frequency>
```

```
SENSe<Ch>:FREQuency:SPAN?
```

<i>Description</i>	Sets or reads out the stimulus span value of the sweep range for linear or logarithmic sweep type (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the stimulus span value from 0 to 1.2997E9 (Planar – TR1300/1), from 0 to 4.79998E9 (TR5048), from 0 to 2.99998E9 (TR7530).
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1.2997E9 (Planar – TR1300/1), 4.79998E9 (TR5048), 2.99998E9 (TR7530).
<i>Equivalent Softkeys</i>	Stimulus > Span

SENS:FREQ:STAR

```
SENSe<Ch>:FREQuency:STARt <frequency>
```

```
SENSe<Ch>:FREQuency:STARt?
```

<i>Description</i>	Sets or reads out the stimulus start value of the sweep range for linear or logarithmic sweep type (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the stimulus start value from 3E5 to 1.3E9 (Planar – TR1300/1), from 2E4 to 4.8E9 (TR5048), from 2E4 to 3.0E9 (TR7530).
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	3E5 (Planar – TR1300/1), 2E4 (TR5048), 2E4 (TR7530).
<i>Equivalent Softkeys</i>	Stimulus > Start

SENS:FREQ: STOP

```
SENSe<Ch>:FREQuency:STOP <frequency>
```

```
SENSe<Ch>:FREQuency:STOP?
```

<i>Description</i>	Sets or reads out the stimulus stop value of the sweep range for linear or logarithmic sweep type (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<frequency> the stimulus stop value form 3E5 to 1.3E9 (Planar – TR1300/1), from 2E4 to 4.8E9 (TR5048), from 2E4 to 3.0E9 (TR7530).
<i>Unit</i>	Hz (Hertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1.3E9 (Planar – TR1300/1), 4.8E9 (TR5048), 3.0E9 (TR7530).
<i>Equivalent Softkeys</i>	Stimulus > Stop

SENS:OFFS

```
SENSe<Ch>:OFFSet[:STATe] {ON|OFF|1|0}
```

```
SENSe<Ch>:OFFSet[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the frequency offset feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} frequency offset ON {OFF 0} frequency offset OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Freq. Offset

SENS:OFFS:ADJ

```
SENSe<Ch>:OFFSet:ADJust[:STATe] {ON|OFF|1|0}
```

```
SENSe<Ch>:OFFSet:ADJust[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the frequency offset adjust feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} frequency offset adjust ON {OFF 0} frequency offset adjust OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Offset Adjust

SENS:OFFS:ADJ:CONT

```
SENSe<Ch>:OFFSet:ADJust:CONTInuous[:STATe] {ON|OFF|1|0}
```

```
SENSe<Ch>:OFFSet:ADJust:CONTInuous[:STATe]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the continuous frequency offset adjust feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	{ON 1} continuous frequency offset adjust ON {OFF 0} continuous frequency offset adjust OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Contin. Adjust

SENS:OFFS:ADJ:CONT:PER

```
SENSe<Ch>:OFFSet:ADJust:CONTinuous:PERiod <numeric>
```

```
SENSe<Ch>:OFFSet:ADJust:CONTinuous:PERiod?
```

<i>Description</i>	Sets or reads out the period for continuous offset adjust feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the period for continuous offset adjust feature: 1 - 3 sec, 2 - 10 sec, 3 - 30 sec, 4 - 100 sec, 5 - 300 sec,
<i>Unit</i>	n/a
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	3
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Adjust Period

SENS:OFFS:ADJ:EXEC

SENSe<Ch>:OFFSet:ADJust:EXECute

<i>Description</i>	Executes the offset adjust when the frequency offset feature is ON (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Adjust Immediate

SENS:OFFS:ADJ:PORT

```
SENSe<Ch>:OFFSet:ADJust:PORT <numeric>
```

```
SENSe<Ch>:OFFSet:ADJust:PORT?
```

<i>Description</i>	Sets or reads out the port number for frequency offset adjust feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> port number 1 to 2;
<i>Unit</i>	n/a
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Select Port

SENS:OFFS:ADJ:VAL

```
SENSe<Ch>:OFFSet:ADJust:VALue <hertz>
```

```
SENSe<Ch>:OFFSet:ADJust:VALue?
```

<i>Description</i>	Sets or reads out the frequency offset adjust value of the offset adjust feature (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<hertz> adjust value of the offset adjust feature from -5E5 to 5E5;
<i>Unit</i>	Hz
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Offset Adjust > Adjust Value

SENS:OFFS:PORT:DATA?

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DATA?
```

<i>Description</i>	<p>Reads out the array of the frequency points of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (query only). The array size is N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p><numeric n> the frequency value at the n-th point</p>
<i>Target</i>	<p>Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }</p>
<i>Query Response</i>	<p><numeric 1>, <numeric 2>, ...<numeric N></p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	<p>FORM:DATA SENS:OFFS:STAT SENS:OFFS:TYPE</p>
<i>Equivalent Softkeys</i>	None

SENS:OFFS:PORT:DIV

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DIVisor <numeric>
```

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DIVisor?
```

<i>Description</i>	Sets or reads out the basic frequency range divisor of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<numeric> divisor from 1 to 1000;
<i>Unit</i>	n/a
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Related Commands</i>	SENS:OFFS:STAT SENS:OFFS:TYPE
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Port n > Divider

SENS:OFFS:PORT:MULT

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:MULTiplier <numeric>
```

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:MULTiplier?
```

<i>Description</i>	Sets or reads out the basic frequency range multiplier of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<numeric> multiplier from -1000 to 1000;
<i>Unit</i>	n/a
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Related Commands</i>	SENS:OFFS:STAT SENS:OFFS:TYPE
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Port n > Multiplier

SENS:OFFS:PORT:OFFS

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:OFFSet <hertz>
```

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:OFFSet?
```

<i>Description</i>	Sets or reads out the basic frequency range offset of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Parameter</i>	<hertz> offset from -1e12 to 1e12;
<i>Unit</i>	Hz
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Related Commands</i>	SENS:OFFS:STAT SENS:OFFS:TYPE
<i>Equivalent Softkeys</i>	Stimulus > Freq.Offset > Port n > Offset

SENS:OFFS:PORT:STAR

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STARt <hertz>
```

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STARt?
```

<i>Description</i>	Sets or reads out the frequency sweep start of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Unit</i>	Hz
<i>Query Response</i>	<numeric>
<i>Related Commands</i>	SENS:OFFS:STAT SENS:OFFS:TYPE
<i>Equivalent Softkeys</i>	Stimulus > Freq. Offset > Port n > Start

SENS:OFFS:PORT:STOP

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STOP <hertz>
```

```
SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STOP?
```

<i>Description</i>	Sets or reads out the frequency sweep stop of port <Pt> when the frequency offset feature is ON and offset type is "PORT" (command/query).
<i>Target</i>	Port <Pt> of channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 } <Pt>={ [1] 2 }
<i>Unit</i>	Hz
<i>Query Response</i>	<numeric>
<i>Related Commands</i>	SENS:OFFS:STAT SENS:OFFS:TYPE
<i>Equivalent Softkeys</i>	Stimulus > Freq.Offset > Port n > Stop

SENS:ROSC:SOUR

```
SENSe<Ch>:ROSCillator:SOURce <char>
```

```
SENSe<Ch>:ROSCillator:SOURce?
```

<i>Description</i>	Sets or reads out the internal or external source of the reference frequency of 10 MHz (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> choose from: INTernal : Internal source of the reference frequency EXTernal : External source of the reference frequency
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ INT EXT }
<i>Preset Value</i>	INT
<i>Equivalent Softkeys</i>	System > Misc Setup > Ref Source

SENS:SEGM:DATA

```
SENSe<Ch>:SEGMENT:DATA <numeric list>
```

```
SENSe<Ch>:SEGMENT:DATA?
```

<i>Description</i>	<p>Sets or reads out the array of the segment sweep table (command/query).</p> <p>The array has the following format:</p> <pre>{<Buf>, <Flag1>, <Flag2>, <Flag3>, <Flag4>, <Flag5>, <N>, <Start 1>, <Stop 1>, <NOP 1> [, <IFBW 1>] [, <Pow 1>] [, <Del 1>] [, <Time 1>], <Start 2>, <Stop 2>, <NOP 2> [, <IFBW 2>] [, <Pow 2>] [, <Del 2>] [, <Time 2>], ... <StartN>, <StopN>, <NOP N> [, <IFBW N>] [, <Pow N>] [, <Del N>] [, <TimeN>]}</pre> <p><Buf> Always 5, <Flag1> Stimulus start setting (0 – start/stop, 1 – center/span), <Flag2> Setting of the <IFBW> field (0 – disabled, 1 – enabled), <Flag3> Setting of the <Pow> field (0 – disabled, 1 – enabled), <Flag4> Setting of the field (0 – disabled, 1 – enabled), <Flag5> Setting of the <Time> field (0 – disabled, 1 – enabled), <N> Number of segments, <Start n> Start value of the n-th segment, <Stop n> Stop value of the n-th segment, <NOP n> Number of points of the n-th segment, <IFBW n> IF bandwidth of the n-th segment (if enabled), <Pow n> Power of the n-th segment (if enabled), <Del n> Measurement delay of the n-th segment (if enabled), <Time n> Reserved for future use (if enabled)</p>
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<pre><numeric 1>, <numeric 2>, ...<numeric 7+M×N></pre> <p>Where, N – the number of the segments, M – depends on the values of the flags $M = 3 + \text{<Flag2>} + \text{<Flag3>} + \text{<Flag4>} + \text{<Flag5>}$</p>
<i>Equivalent Softkeys</i>	Stimulus > Segment Table

SENS:SWE:POIN

```
SENSe<Ch>:SWEep:POINts <numeric>
```

```
SENSe<Ch>:SWEep:POINts?
```

<i>Description</i>	Sets or reads out the number of measurement points (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the number of measurement points from 2 to 16001 (Planar – TR1300/1), from 2 to 100001 (TR5048), from 2 to 100001 (TR7530).
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	201
<i>Equivalent Softkeys</i>	Stimulus > Points

SENS:SWE:POIN:TIME

```
SENSe<Ch>:SWEep:POINt:TIME <time>
```

```
SENSe<Ch>:SWEep:POINt:TIME?
```

<i>Description</i>	Sets or reads out the delay before measurement in each measurement point (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<time> the measurement delay value from 0 to 0.3
<i>Resolution</i>	5E-6
<i>Unit</i>	s (second)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Meas Delay

SENS:SWE:TYPE

```
SENSe<Ch>:SWEep:TYPE <char>
```

```
SENSe<Ch>:SWEep:TYPE?
```

<i>Description</i>	Sets or reads out the sweep type (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> specifies the sweep type:</p> <p>LINear : Linear frequency sweep</p> <p>LOGarithmic : Logarithmic frequency sweep</p> <p>SEGment : Segment frequency sweep</p> <p>POWer : Power sweep</p> <p>VVM : Vector Voltmeter mode</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ LIN LOG SEGM POW }
<i>Preset Value</i>	LIN
<i>Equivalent Softkeys</i>	Stimulus > Sweep Type

SENS:VVM:TYPE

SENSe<Ch>:VVM:TYPE <char>

SENSe<Ch>:VVM:TYPE?

<i>Description</i>	Sets or reads out the smeasured parameter in Vector Voltmeter mode.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<p><char> choose from:</p> <p>S11 : Reflection parameter</p> <p>S21 : Transmission parameter</p> <p>A/B : The ratio of the receivers signals A to B, an external frequency generator is used.</p> <p>B/A : The ratio of the receivers signals B to A, an external frequency generator is used.</p>
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	{S11 S21 A/B B/A}
<i>Preset Value</i>	S11
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Measurement > {S11 S21 A/B B/A}

SENS:VVM:FORM

SENSe<Ch>:VVM:FORMat <char>

SENSe<Ch>:VVM:FORMat?

<i>Description</i>	Sets or reads out the data format in Vector Voltmeter mode.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<char> choose from: LOGP : Log magnitude and phase LINP : Linear magnitude and phase SWR : Voltage standing wave ratio IMP : Impedance value
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	{ LOGP LINP SWR IMP }
<i>Preset Value</i>	LOGP
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Format > {LOGP LINP SWR IMP}

SENS:VVM:FREQ

```
SENSe<Ch>:VVM:FREQuency[:CW] <numeric>
```

```
SENSe<Ch>:VVM:FREQuency[:CW]?
```

<i>Description</i>	Sets or reads out the operating frequency in the Vector Voltmeter mode.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> the frequency value.
<i>Unit</i>	Hz (Hertz)
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Frequency

SENS:VVM:DATA?

SENSe<Ch>:VVM:DATA?

<i>Description</i>	<p>Reads out the value of the vector voltmeter measurement (query only). If a reference value is specified, the return value is read relatively to it.</p> <p>The data include 2 elements:</p> <p><numeric 1> Magnitude value, depending on the data format or SWR value;</p> <p><numeric 2> Phase value or zero in SWR format.</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>
<i>Related Commands</i>	<p>SENSe<Ch>:VVM:FORMat</p> <p>SENSe<Ch>:VVM:REFerence:MEMorize</p> <p>SENSe<Ch>:VVM:REFerence:CLEar</p>
<i>Equivalent Softkeys</i>	None

SENS:VVM:REF:CLE

SENSe<Ch>:VVM:REFerence:CLEar

<i>Description</i>	Clears the reference value of the vector voltmeter .
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Clear Reference

SENS:VVM:REF:MEM

SENSe<Ch>:VVM:REFerence:MEMorize

<i>Description</i>	Memorizes the reference value of the vector voltmeter .
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Save Reference

SENS:VVM:REF:DATA?

SENSe<Ch>:VVM:REFerence:DATA?

<i>Description</i>	Reads out the reference value of the vector voltmeter measurement (query only). The data include 2 elements: <numeric 1> Magnitude value, depending on the data format or SWR value; <numeric 2> Phase value or zero in SWR format.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Query Response</i>	<numeric 1>, <numeric 2>
<i>Related Commands</i>	SENSe<Ch>:VVM:FORMat SENSe<Ch>:VVM:REFerence:MEMorize SENSe<Ch>:VVM:REFerence:CLEar
<i>Equivalent Softkeys</i>	None

SENS:VVM:TABLE:CLE

SENSe<Ch>:VVM:TABLE:CLEAr

<i>Description</i>	Clears all data from the vector voltmeter measurement table.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Related Commands</i>	SENSe<Ch>:VVM:TABLE:MEMorize SENSe<Ch>:VVM:TABLE:INSert SENSe<Ch>:VVM:TABLE:REMOve
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Data Table > Clear Table

SENS:VVM:TABLE:MEM

SENSe<Ch>:VVM:TABLE:MEMorize

<i>Description</i>	Saves measurements of the vector voltmeter to the table.
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Related Commands</i>	SENSe<Ch>:VVM:TABLE:MEMorize SENSe<Ch>:VVM:TABLE:INSert SENSe<Ch>:VVM:TABLE:REMOve
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Data Table > Add Data

SENS:VVM:TABL:INS

SENSe<Ch>:VVM:TABLE:MEMorize

<i>Description</i>	<p>Inserts a row of data into the table with the current measurement of a vector voltmeter</p> <p><numeric> The row number in the table where new data will be inserted.</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Related Commands</i>	<p>SENSe<Ch>:VVM:TABLE:MEMorize SENSe<Ch>:VVM:TABLE:INSert SENSe<Ch>:VVM:TABLE:REMOve</p>
<i>Equivalent Softkeys</i>	<p>Stimulus > Vector Voltmeter > Data Table > Insert Data</p>

SENS:VVM:TABL:REM

SENSe<Ch>:VVM:TABLE:REMOve <numeric>

<i>Description</i>	<p>Removes a table row with a vector voltmeter measurement index</p> <p><numeric> The row number in the table which will be removed.</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Related Commands</i>	<p>SENSe<Ch>:VVM:TABLE:CLEar SENSe<Ch>:VVM:TABLE:MEMorize SENSe<Ch>:VVM:TABLE:INSert</p>
<i>Equivalent Softkeys</i>	<p>Stimulus > Vector Voltmeter > Data Table > Remove Data</p>

SENS:VVM:TABL:SAVE

SENSe<Ch>:VVM:TABLe:SAVE <string>

<i>Description</i>	Saves the table into CSV file.
<i>Target</i>	Channel <Ch>, <Ch>={{1} 2 3 4 5 6 7 8 9}}
<i>Parameter</i>	<string> File name (quoted string)
<i>Notes</i>	If the full path of the file is not specified, the \CSV subdirectory of the main directory will be searched for the file. The file has *.csv extension by default.
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Data Table > Save Table to CSV

SENS:VVM:TABL:DATA?

SENSe<Ch>:VVM:TABLE:DATA

<i>Description</i>	<p>Returns data table of the vector voltmeter (query only). The array has the following format:</p> <pre>{<Flag1>, <N>, <Abs.Mag 1>, <Abs.Phase 1> [,<Rel.Mag 1>] [,<Rel.Phase 1>], ... <Abs.Mag N>, <Abs.Phase N> [,<Rel.Mag N>] [,<Rel.Phase N>]},</pre> <p>Where: <Flag1> reference value, 0 – is not specified, 1 – is specified, N – rows number in the data table; <Abs.Mag i>, <Abs.Phase i> - measurement data of a vector voltmeter; [,<Rel.Mag i>] [,<Rel.Phase i>] - measurement data relative to the reference value (if specified).</p>
<i>Target</i>	<p>Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }</p>
<i>Query Response</i>	<p><numeric 1>,<numeric 2>,...<numeric 2+N×(2+M×2)> Where: N – number of table rows, M – flag value Flag1</p>
<i>Related Commands</i>	<p>SENSe<Ch>:VVM:TABLE:CLear SENSe<Ch>:VVM:TABLE:MEMorize SENSe<Ch>:VVM:TABLE:REMOve SENSe<Ch>:VVM:TABLE:INSert</p>
<i>Equivalent Softkeys</i>	<p>None</p>

SERV:CHAN:ACT?

SERVice:CHANnel:ACTive?

<i>Description</i>	Reads out the active channel number (query only).
<i>Query Response</i>	<numeric> from 1 to 16
<i>Equivalent Softkeys</i>	None

SERV:CHAN:COUN?

SERVice:CHANnel:COUNT?

<i>Description</i>	Reads out the maximum number of the channels (query only).
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SERV:CHAN:TRAC:ACT?

SERVice:CHANnel<Ch>:TRACe:ACTive?

<i>Description</i>	Read out the active trace number of the channel (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9}
<i>Query Response</i>	<numeric> from 1 to 16
<i>Equivalent Softkeys</i>	None

SERV:CHAN:TRAC:COUN?

SERVice:CHANnel:TRACe:COUNT?

<i>Description</i>	Reads out the maximum number of the traces in the channel (query only).
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SERV:PORT:COUN?

SERVice:PORT:COUNT?

<i>Description</i>	Reads out the number of the ports (query only).
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SERV:SWE:FREQ:MAX?

SERVice:SWEep:FREQuency:MAXimum?

<i>Description</i>	Reads out the upper limit of the measurement frequency (query only).
<i>Query Response</i>	<numeric>
<i>Unit</i>	Hz (Hertz)
<i>Equivalent Softkeys</i>	None

SERV:SWE:FREQ:MIN?

SERVice:SWEep:FREQuency:MINimum?

<i>Description</i>	Reads out the lower frequency of the measurement frequency (query only).
<i>Query Response</i>	<numeric>
<i>Unit</i>	Hz (Hertz)
<i>Equivalent Softkeys</i>	None

SERV:SWE:POIN?

SERVice:SWEep:POINts?

<i>Description</i>	Reads out the maximum number of the measurement points (query only).
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SOUR:POW

```
SOURce<Ch>:POWer[:LEVel][:IMMediate][:AMPLitude] <power>
```

```
SOURce<Ch>:POWer[:LEVel][:IMMediate][:AMPLitude]?
```

<i>Description</i>	Sets or reads out the power level for the frequency sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power level from -55 to +3
<i>Resolution</i>	0.05
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Power > Output Power

SOUR:POW:CENT

```
SOURce<Ch>:POWer:CENTer <power>
```

```
SOURce<Ch>:POWer:CENTer?
```

<i>Description</i>	Sets or reads out the center value of the power sweep range (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power sweep center value from -55 to +3
<i>Resolution</i>	0.025
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	-22.5
<i>Equivalent Softkeys</i>	Stimulus > Center

SOUR:POW:SLOP

```
SOURce<Ch>:POWer[:LEVel]:SLOPe[:DATA] <power>
```

```
SOURce<Ch>:POWer[:LEVel]:SLOPe[:DATA]?
```

<i>Description</i>	Sets or reads out the power slope value for the frequency sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power slope value from -2 to +2
<i>Resolution</i>	0.1
<i>Unit</i>	dB/GHz (decibel/gigahertz)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	Stimulus > Power > Power Slope [dB/GHz]

SOUR:POW:SPAN

```
SOURce<Ch>:POWer:SPAN <power>
```

```
SOURce<Ch>:POWer:SPAN?
```

<i>Description</i>	Sets or reads out the power span for the power sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power sweep span value from 0 to 58
<i>Resolution</i>	0.05
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	58
<i>Equivalent Softkeys</i>	Stimulus > Span

SOUR:POW:STAR

```
SOURce<Ch>:POWer:STARt <power>
```

```
SOURce<Ch>:POWer:STARt?
```

<i>Description</i>	Sets or reads out the power sweep start for the power sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power sweep start from -55 to +3
<i>Resolution</i>	0.05
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	-55
<i>Equivalent Softkeys</i>	Stimulus > Start

SOUR:POW:STOP

```
SOURce<Ch>:POWer:STOP <power>
```

```
SOURce<Ch>:POWer:STOP?
```

<i>Description</i>	Sets or reads out the power sweep stop for the power sweep (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> the power sweep stop from -55 to +3
<i>Resolution</i>	0.05
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	3
<i>Equivalent Softkeys</i>	Stimulus > Stop

STAT:OPER?

STATus:OPERation[:EVENT]?

<i>Description</i>	Reads out the value of the Operation Status Event Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:OPER:COND?

STATus:OPERation:CONDition?

<i>Description</i>	Reads out the value of the Operation Status Condition Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:OPER:ENAB

```
STATus:OPERation:ENABle <numeric>
```

```
STATus:OPERation:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Operation Status Enable Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:OPER:NTR

STATus:OPERation:NTRansition <numeric>

STATus:OPERation:NTRansition?

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Operation Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:OPER:PTR

```
STATus:OPERation:PTRansition <numeric>
```

```
STATus:OPERation:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Operation Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:PRES

```
STATus:PRESet
```

<i>Description</i>	Resets all the status registers to the factory settings (no query).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:COND?

```
STATus:QUEStionable:CONDition?
```

<i>Description</i>	Reads out the value of the Questionable Status Condition Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:ENAB

```
STATus:QUEStionable:ENABle <numeric>
```

```
STATus:QUEStionable:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Questionable Status Enable Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:CHAN:COND?

```
STATus:QUESTionable:LIMit:CHANnel<Ch>:CONDition?
```

<i>Description</i>	Reads out the value of the Questionable Limit Channel Status Condition Register (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:CHAN:ENAB

```
STATus:QUESTionable:LIMit:CHANnel<Ch>:ENABle <numeric>
```

```
STATus:QUESTionable:LIMit:CHANnel<Ch>:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Questionable Limit Channel Status Enable Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:CHAN:NTR

```
STATus:QUEStionable:LIMit:CHANnel<Ch>:NTRansition <numeric>
```

```
STATus:QUEStionable:LIMit:CHANnel<Ch>:NTRansition?
```

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Questionable Limit Channel Status Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:CHAN:PTR

```
STATus:QUEStionable:LIMit:CHANnel<Ch>:PTRansition <numeric>
```

```
STATus:QUEStionable:LIMit:CHANnel<Ch>:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Questionable Limit Channel Status Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:CHAN?

```
STATus:QUESTionable:LIMit:CHANnel<Ch>[:EVENT]?
```

<i>Description</i>	Reads out the value of the Questionable Limit Channel Status Event Register (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:COND?

```
STATus:QUESTionable:LIMit:CONDition?
```

<i>Description</i>	Reads out the value of the Questionable Limit Status Condition Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:ENAB

```
STATus:QUEStionable:LIMit:ENABle <numeric>
```

```
STATus:QUEStionable:LIMit:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Questionable Limit Status Enable Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:NTR

```
STATus:QUEStionable:LIMit:NTRansition <numeric>
```

```
STATus:QUEStionable:LIMit:NTRansition?
```

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Questionable Limit Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM:PTR

```
STATus:QUEStionable:LIMit:PTRansition <numeric>
```

```
STATus:QUEStionable:LIMit:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Questionable Limit Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:QUES:LIM?

```
STATus:QUEStionable:LIMit[:EVENT]?
```

<i>Description</i>	Reads out the value of the Questionable Limit Status Event Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:NTR

```
STATus:QUEStionable:NTRansition <numeric>
```

```
STATus:QUEStionable:NTRansition?
```

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Questionable Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:PTR

```
STATus:QUEStionable:PTRansition <numeric>
```

```
STATus:QUEStionable:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Questionable Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:CHAN:COND?

```
STATus:QUESTionable:RLIMit:CHANnel<Ch>:CONDition?
```

<i>Description</i>	Reads out the value of the Questionable Ripple Limit Channel Status Condition Register (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:CHAN:ENAB

```
STATus:QUESTionable:RLIMit:CHANnel<Ch>:ENABle <numeric>
```

```
STATus:QUESTionable:RLIMit:CHANnel<Ch>:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Questionable Ripple Limit Channel Status Enable Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:CHAN:NTR

```
STATus:QUEStionable:RLIMit:CHANnel<Ch>:NTRansition <numeric>
```

```
STATus:QUEStionable:RLIMit:CHANnel<Ch>:NTRansition?
```

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Questionable Ripple Limit Channel Status Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:CHAN:PTR

```
STATus:QUEStionable:RLIMit:CHANnel<Ch>:PTRansition <numeric>
```

```
STATus:QUEStionable:RLIMit:CHANnel<Ch>:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Questionable Ripple Limit Channel Status Register (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:CHAN?

```
STATus:QUEStionable:RLIMit:CHANnel<Ch>[:EVENT]?
```

<i>Description</i>	Reads out the value of the Questionable Ripple Limit Channel Status Event Register (query only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1] 2 3 4 5 6 7 8 9 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:COND?

```
STATus:QUESTionable:RLIMit:CONDition?
```

<i>Description</i>	Reads out the value of the Questionable Ripple Limit Status Condition Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:ENAB

```
STATus:QUESTionable:RLIMit:ENABle <numeric>
```

```
STATus:QUESTionable:RLIMit:ENABle?
```

<i>Description</i>	Sets or reads out the value of the Questionable Ripple Limit Status Enable Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:NTR

```
STATus:QUEStionable:RLIMit:NTRansition <numeric>
```

```
STATus:QUEStionable:RLIMit:NTRansition?
```

<i>Description</i>	Sets or reads out the value of the Negative transition filter of the Questionable Ripple Limit Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	0
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM:PTR

```
STATus:QUEStionable:RLIMit:PTRansition <numeric>
```

```
STATus:QUEStionable:RLIMit:PTRansition?
```

<i>Description</i>	Sets or reads out the value of the Positive transition filter of the Questionable Ripple Limit Status Register (command/query).
<i>Target</i>	Status Reporting System
<i>Parameter</i>	<numeric> from 0 to 65535
<i>Out of Range</i>	Bit-to-bit AND with numeric 65535
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	65535
<i>Equivalent Softkeys</i>	None

STAT:QUES:RLIM?

```
STATus:QUEStionable:RLIMit[:EVENT]?
```

<i>Description</i>	Reads out the value of the Questionable Ripple Limit Status Event Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

STAT:QUES?

STATus:QUEStionable[:EVENT]?

<i>Description</i>	Reads out the value of the Questionable Status Event Register (query only).
<i>Target</i>	Status Reporting System
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SYST:COMM:ECAL:TEMP:SENS?

SYSTem:COMMunicate:ECAL:TEMPerature:SENSor?

<i>Description</i>	Reads out the temperature of the AutoCal module connected to the Analyzer. (query only)
<i>Target</i>	AutoCal module
<i>Unit</i>	°C
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SYST:CORR

```
SYSTem:CORRection[:StAtE] {ON|OFF|1|0}
```

```
SYSTem:CORRection[:StAtE]?
```

<i>Description</i>	Sets or reads out the ON/OFF state of the system error correction (command/query).
<i>Parameter</i>	{ON 1} System error correction ON {OFF 0} System error correction OFF
<i>Query Response</i>	{0 1}
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	System > Misc Setup > System Correction

SYST:DATE

```
SYSTem:DATE <numeric 1>,<numeric 2>,<numeric 3>
```

```
SYSTem:DATE?
```

<i>Description</i>	Sets or reads out the current date (command/query).
<i>Parameter</i>	<numeric 1> year from 1900 to 2100; <numeric 2> month from 1 to 12; <numeric 3> day from 1 to 31.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Equivalent Softkeys</i>	None

SYST:ERR?

SYSTem:ERRor[:NEXT]?

<i>Description</i>	Reads out the error message of the error occurred when executing the SCPI commands, from the FIFO (First In First Out) error queue stored in the instrument. The read out error is deleted from the error queue. The *CLS command clears the error queue. The maximum size of the queue is 100 messages (query only).
<i>Query Response</i>	<p><numeric>, <string> Where:</p> <p><numeric> error code <string> error message</p> <p>If there is no error in the queue, "0, No error" is read out.</p>
<i>Equivalent Softkeys</i>	None

SYST:LOC

SYSTem:LOCal

<i>Description</i>	Sets the instrument to the local operation mode, when all the keys on the front panel, mouse and the touch screen are active (no query).
<i>Related Commands</i>	<p>SYST:REM SYST:RWL</p>
<i>Equivalent Softkeys</i>	None

SYST:PRES

SYSTem:PRESet

<i>Description</i>	Resets the instrument to the factory settings (no query).
<i>Notes</i>	The difference from the *RST: command is that the trigger is set to the Continuous trigger mode.
<i>Related Commands</i>	*RST
<i>Equivalent Softkeys</i>	System > Preset > Apply

SYST: READy?

SYSTem:READy [:STATe] ?

<i>Description</i>	Reads out the ready state of the Analyzer. The state is <i>True</i> when analyzer hardware is connected, powered and the boot process is completed (about 10 sec). (query only)
<i>Query Response</i>	{0 1}
<i>Equivalent Softkeys</i>	None

SYST:REM

SYSTem:REMOte

<i>Description</i>	Sets the instrument to the remote operation mode, when all the keys on the front panel, mouse and the touch screen are not active, except for one key labeled Return to Local. Pushing this button will reset the instrument to the local operation mode (no query).
<i>Related Commands</i>	SYST:LOC SYST:RWL
<i>Equivalent Softkeys</i>	None

SYST:RWL

SYSTem:RWLock

<i>Description</i>	Sets the instrument to the remote operation mode, when all the keys on the front panel, mouse and the touch screen are not active. Only SYST:LOC or SYST:REM command can release this remote operation mode (no query).
<i>Related Commands</i>	SYST:LOC SYST:REM
<i>Equivalent Softkeys</i>	None

SYST:TEMP:SENS?

SYSTem:TEMPerature:SENSor? <numeric>

<i>Description</i>	Reads out the specified sensor temperature inside the Analyzer. (query only)
<i>Target</i>	Analyzer
<i>Parameter</i>	<numeric> : Sensor number
<i>Unit</i>	°C
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	None

SYST:TIME

```
SYSTem:TIME <numeric 1>,<numeric 2>,<numeric 3>
```

```
SYSTem:TIME?
```

<i>Description</i>	Sets or reads out the current time (command/query).
<i>Parameter</i>	<p><numeric 1> hours from 0 to 23;</p> <p><numeric 2> minutes from 0 to 59;</p> <p><numeric 3> seconds from 0 to 59.</p>
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Equivalent Softkeys</i>	None

SYST:TERM

```
SYSTem:TERMIte
```

<i>Description</i>	Terminates the analyzer software. (no query)
<i>Equivalent Softkeys</i>	System > Exit > Ok

TRIG

TRIGger[:SEQuence][:IMMediate]

<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the <i>External</i>). If the trigger source is set to <i>External</i>, an error occurs and the command is ignored.</p> <p>If the instrument is not in the waiting for a trigger state (sweep is in progress or all the channels are set to <i>Hold</i>), an error occurs and the command is ignored.</p> <p>The command is completed before the end of the sweep. (no query)</p>
<i>Related Commands</i>	<p>TRIG:SOUR INIT:CONT INIT:IMM</p>
<i>Equivalent Softkeys</i>	None

TRIG:SING

TRIGger[:SEQuence]:SINGle

<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the <i>External</i>). If the trigger source is set to <i>External</i>, an error occurs and the command is ignored.</p> <p>If the instrument is not in the waiting for a trigger state (sweep is in progress or all the channels are set to <i>Hold</i>), an error occurs and the command is ignored.</p> <p>The command is not completed before the end of the sweep (waiting for the completion of the sweep of all the channels). (no query)</p>
<i>Related Commands</i>	<p>TRIG:SOUR INIT:CONT INIT:IMM</p>
<i>Equivalent Softkeys</i>	None

TRIG:SOUR

```
TRIGger[:SEQuence]:SOURce <char>
```

```
TRIGger[:SEQuence]:SOURce?
```

<i>Description</i>	Sets or reads out the sweep trigger source (command/query).
<i>Parameter</i>	<p><char> choose from:</p> <p>INTernal : Internal</p> <p>EXTernal : External</p> <p>MANual : Manual</p> <p>BUS : Bus</p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ INT EXT MAN BUS }
<i>Preset Value</i>	INT
<i>Related Commands</i>	<p>TRIG</p> <p>TRIG:SING</p> <p>*TRG</p>
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Source > { Internal External Bus }

TRIG:STAT?

TRIGger [:SEQuence] :STATus?

<i>Description</i>	Reads out the the current state of the analyzer. (query only)
<i>Parameter</i>	HOLD scanning is stopped MEASure is scanning WAIT is scanning
<i>Equivalent Softkeys</i>	None

TRIG:WAIT

TRIGger[:SEquence]:WAIT <char>

<i>Description</i>	<p>Delays the execution by the analyzer of the next command till the specified state of the analyzer has been reached (see options below).</p> <p>A query that follows the TRIG:WAIT command blocks the execution of the user program till the specified state of the analyzer has been reached.</p> <p>The command can be used to wait for the end of the sweep initiated by the commands TRIG, *TRG or initiated by the external trigger signal. If the <i>continuous initiation mode</i> is turned ON by the command INIT:CONT ON, then the parameter of the command must be WAIT, otherwise HOLD.</p> <p>(no query)</p>
<i>Parameter</i>	<p><char> choose from:</p> <p>HOLD : Waits for the <i>Hold</i> state</p> <p>MEASure : Waits for the <i>Measure</i> state</p> <p>WAIT : Waits for the <i>Trigger Waiting</i> state</p>
<i>Related Commands</i>	<p>TRIG, *TRG</p> <p>TRIG:SOUR EXT</p>
<i>Equivalent Softkeys</i>	<p>None</p>

VVM:RES

VVM[:DISPlay]:RESolution <numeric>

VVM[:DISPlay]:RESolution?

<i>Description</i>	Sets or reads out number of indication symbols in the vector voltmeter mode
<i>Parameter</i>	Number of significant symbols from 3 to 12
<i>Query Response</i>	<numeric>
<i>Preset value</i>	5
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Properties > Response Digits

VVM:FONT

VVM[:DISPlay]:FONT[:SIZE] <numeric>

VVM[:DISPlay]:FONT[:SIZE]?

<i>Description</i>	Sets or reads out the font size for indication in the vector voltmeter mode
<i>Parameter</i>	Font size from 3 to 12
<i>Query Response</i>	<numeric>
<i>Preset value</i>	36
<i>Equivalent Softkeys</i>	Stimulus > Vector Voltmeter > Properties > Font Size

5 Programming Tips

This section gives recommendations for programming in certain specific situations.

5.1 Program Sweep Initiation and Waiting

The program sweep initiation and waiting for its completion the most simple method can be implemented by using commands `TRIG:SING` and `*OPC?`. The command `TRIG:SING` gerates a trigger signal and starts sweeping under the following conditions:

- The program trigger source is selected by command `TRIG:SOUR BUS`;
- The analyzer should be in the *trigger waiting* state, otherwise (the analyzer is sweeping or analyzer is in the *Hold* state) an error occurs and the command is ignored.

The transition of the analyzer to the *trigger waiting* state depends on the state of the *continuous initiation mode*, which is set by command `INIT:CONT`. Provided that the *continuous initiation mode* is ON, the analyzer automatically transits to the *trigger waiting* state when the program trigger source have been selected, and then each time at the end of a sweep. Provided that the *continuous initiation mode* is OFF, the analyzer transits to the *trigger waiting* state for single time upon receiving the command `INIT`.

The command `TRIG:SING` remains pending until the end of sweep. This allows use the `*OPC?` query for the waiting the end of sweep.

Example 1. Start sweeping in all channels and waits for its completion. The channels sweep one by one in turn. The *continuous initiation mode* is enabled by default.

<code>TRIG:SOUR BUS</code>	Selects the program trigger source and puts the analyzer to the <i>trigger waiting</i> state
<code><loop>:</code>	
<code>TRIG:SING</code>	Starts sweep
<code>*OPC?</code>	Waits for the end of the sweep
<code>...</code>	

After the sweep completion the analyser returns to the *trigger waiting* state, and then the next sweep initiation command can be sended.

Example 2. Start sweeping in one channel and waits for its completion, then start sweeping in another channel and waits for its completion. The channel number must be set to 2.

TRIG:SOUR BUS	Selects the program trigger source
INIT1:CONT OFF	Puts channel 1 to the hold state
INIT2:CONT OFF	Puts channel 2 to the hold state
<loop>:	
INIT1	Puts channel 1 to the trigger waiting state
TRIG:SING	Starts sweep in channel 1
*OPC?	Waits for the end of the sweep
...	
INIT2	Puts channel 2 to the trigger waiting state
TRIG:SING	Starts sweep in channel 2
*OPC?	Waits for the end of the sweep
...	

After completion of the sweep on one channel the analyser returns to the *hold* state, then the sweep initiation for another channel is available.

5.2 External Trigger

If trigger source is set to the *External* by the command `TRIG:SOUR EXT`, the sweep starts at signal arrival on the external trigger input.

At the moment of a trigger signal arrival the Analyzer must be in the trigger waiting state, otherwise the signal is ignored but no error is detected.

As a rule the hardware trigger output is used to determine the end of the sweep. If there is a need to determine programmatically the end of the sweep, the `TRIG:WAIT` command can be used. Using program way to determine the end of the sweep requires the *continuous initiation mode* turned OFF.

Example 3. Starts sweep at the signal on the external trigger input. Programmatically waits for its completion. The *continuous initiation mode* must be turned OFF.

TRIG:SOUR EXT	Selects the external trigger source
INIT:CONT OFF	Puts channel 1 to the hold state
<loop>:	

INIT	Puts channel 1 to the <i>trigger waiting</i> state. The sweep starts at external trigger signal
TRIG:WAIT HOLD	Waits for the end of the sweep
*OPC?	Any query is required
...	

After completion of the sweep the analyser returns to the *hold* state, then the next sweep initiation is available.

5.3 Waiting for Calibration Commands

Depending on the sweep settings the calibration commands may have long execution time, as they start sweep and wait it completion. These commands are:

```
SENS:CORR:COLL:XXXX
SENS:CORR:OFFS:COLL:XXXX
SENS:CORR:REC:COLL:XXXX
SENS:CORR:COLL:ECAL:XXXX
SENS:CORR:COLL:ECAL:ORI:EXEC
```

The user program can determine the end of these command using any query, the *OPC? for example.

5.4 VISA Timeout Considerations

When using the query *OPC? for waiting an operation the VISA timeout value must be set to the value no less than the expected sweep time. For example in C/C++:

```
viSetAttribute(instr, VI_ATTR_TMO_VALUE, 5000);
```

5.5 Receiving Data Arrays in Text Form

By default the data from the analyzer is transmitted in a text form. The VISA library has built-in facilities for receiving an array of data from the analyzer. The example assumes that the size of the array is sufficient to receive a number of elements equal to twice the number of points.

An example of receiving a data array in C/C++:

```
double dataArray[1000];
ViUInt32 retCount;
...
```

```
retCount = sizeof(DataArray) / sizeof(double);  
viQueryf(instr, "CALC:DATA:SDAT?\n", "%,#lf", &retCount, Data);  
// retCount now contains the actual number of elements read
```

5.6 Receiving Data Arrays in Binary Form

The transfer of data from the analyzer in binary form reduces the amount of data transferred and the transmission time. The transfer of binary data is supported only by the HiSLIP protocol. To enable binary data transfer, use the `FORM: DATA` command. The list of commands that support the transfer of binary data is given in the description of the `FORM: DATA` command.

Binary data is transmitted as a block having a header followed by data. Block format:

#	8	<Data Size>	<Binary Data>
---	---	-------------	---------------

Where:

– the character '#';

8 – the character '8';

<Data Size> – 8 bytes, the symbolic representation of the number of bytes of binary data.

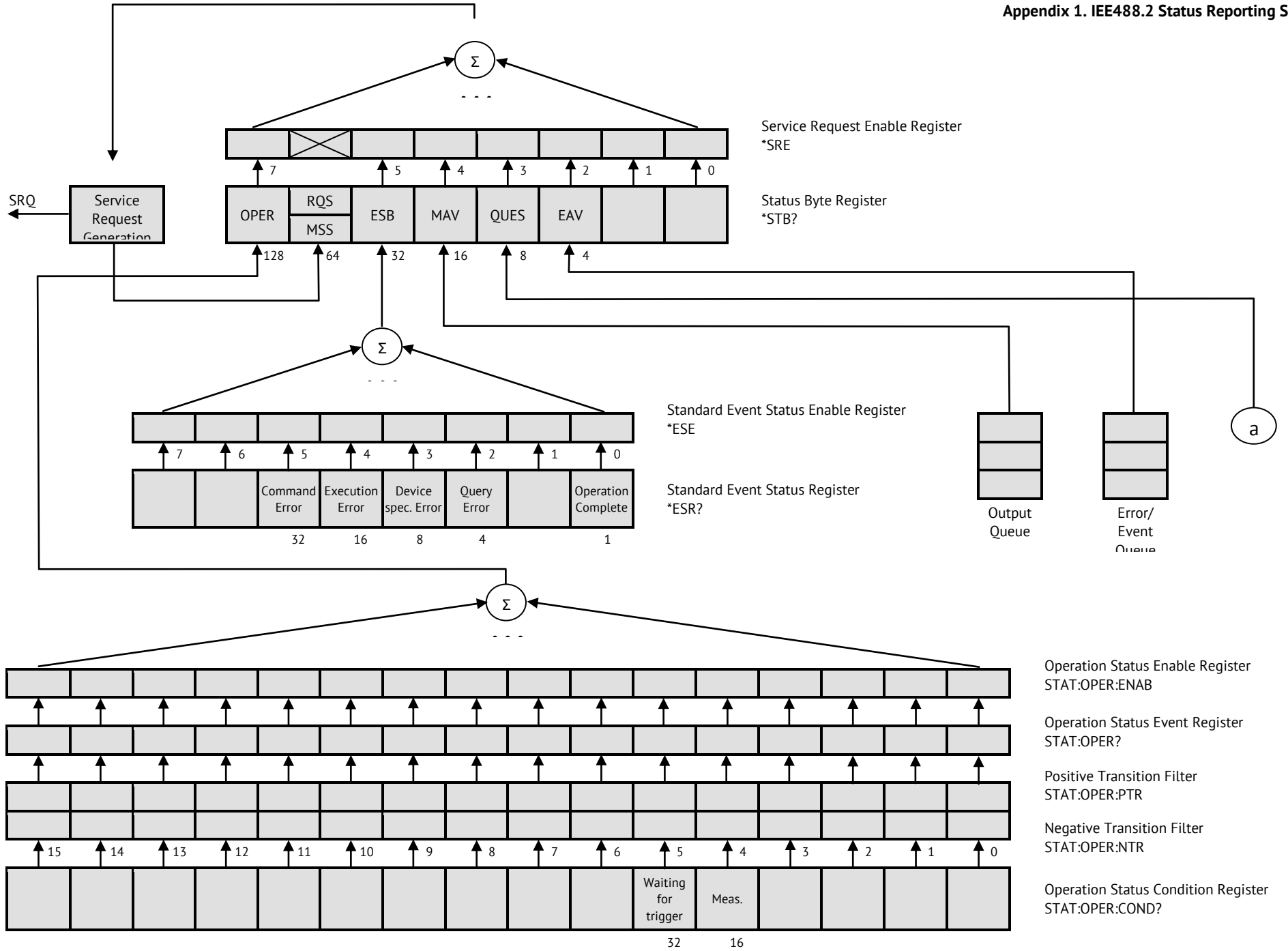
For example:

#800003216<Binary Data>

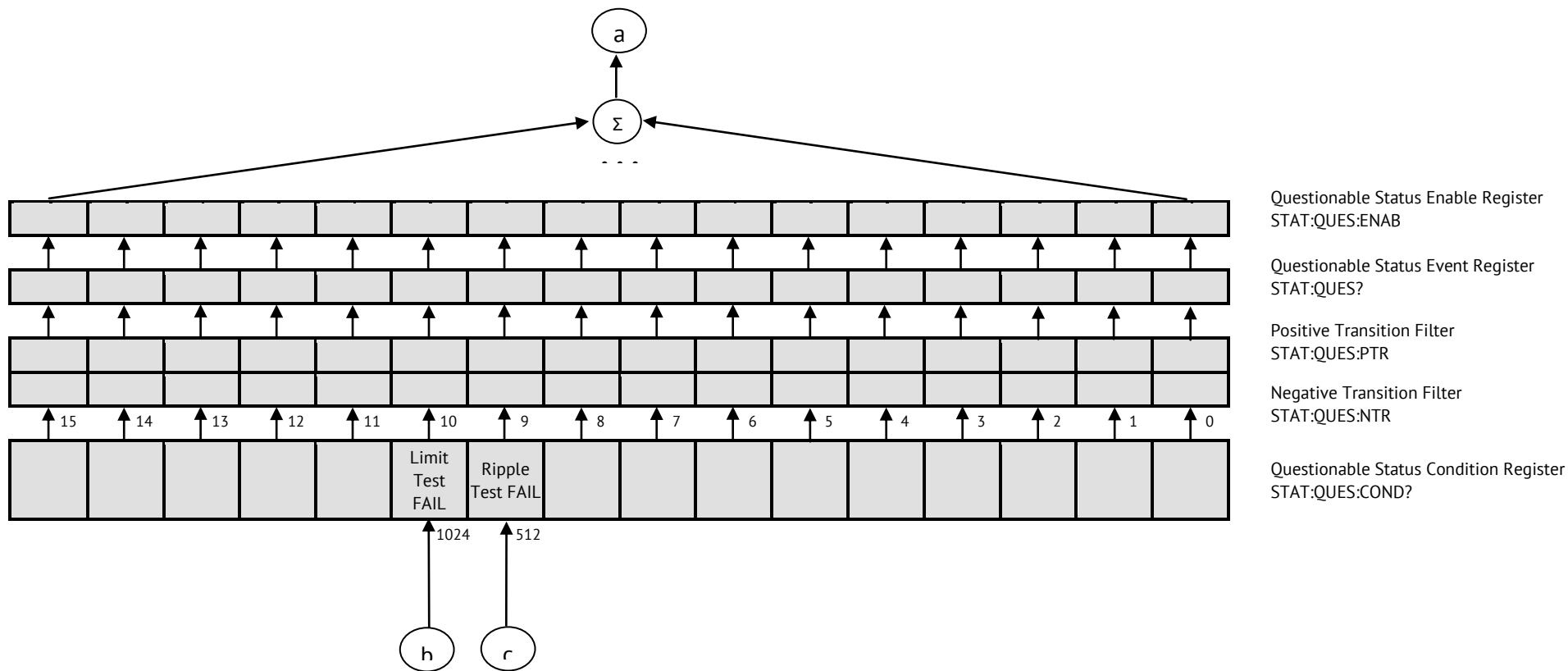
The VISA library has built-in tools for receiving binary data from the analyzer. The example assumes that the size of the array is sufficient to receive a number of elements equal to twice the number of points.

Example of receiving binary data array in C/C ++:

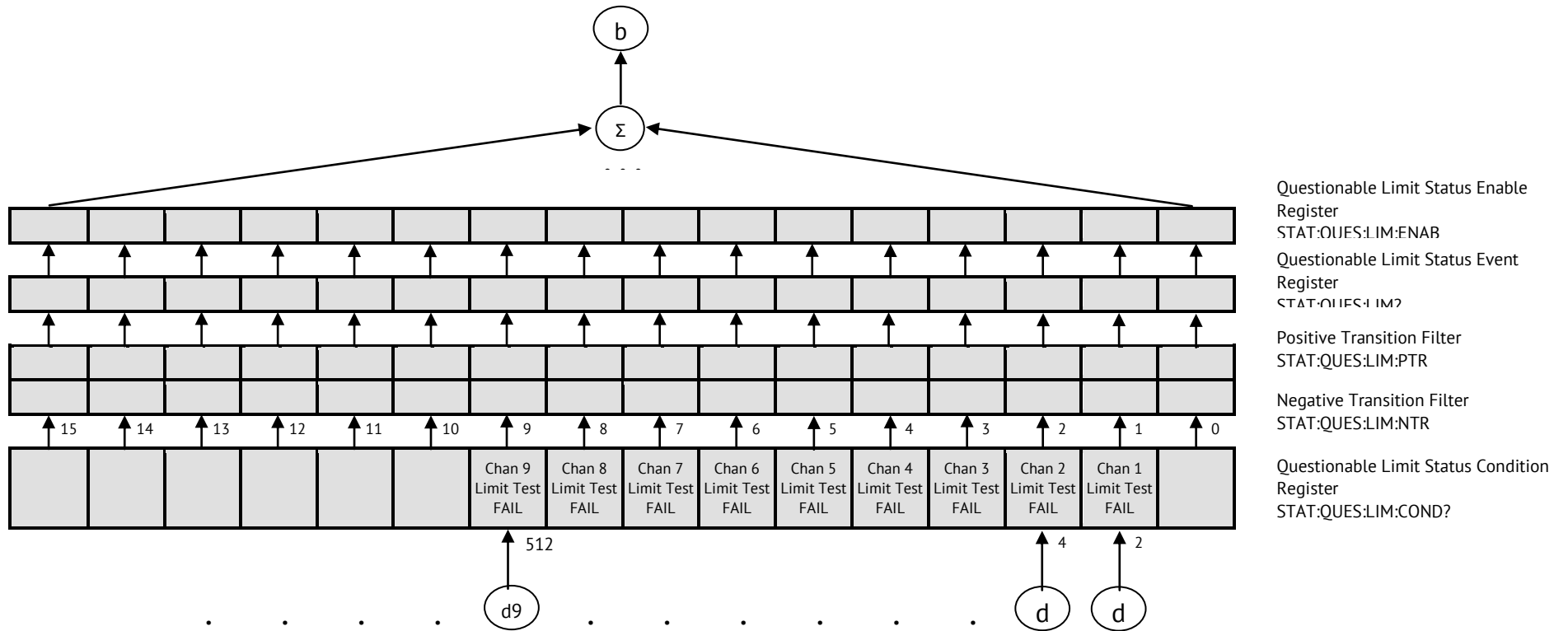
```
double dataArray[1000];
ViUInt32 retCount;
...
viPrintf(instr, "FORM:DATA REAL\n");
retCount = sizeof(DataArray) / sizeof(double);
viQueryf(instr, "CALC:DATA:FDAT?\n", "%#Zb", &retCount, dataArray);
// retCount now contains the actual number of elements read
```

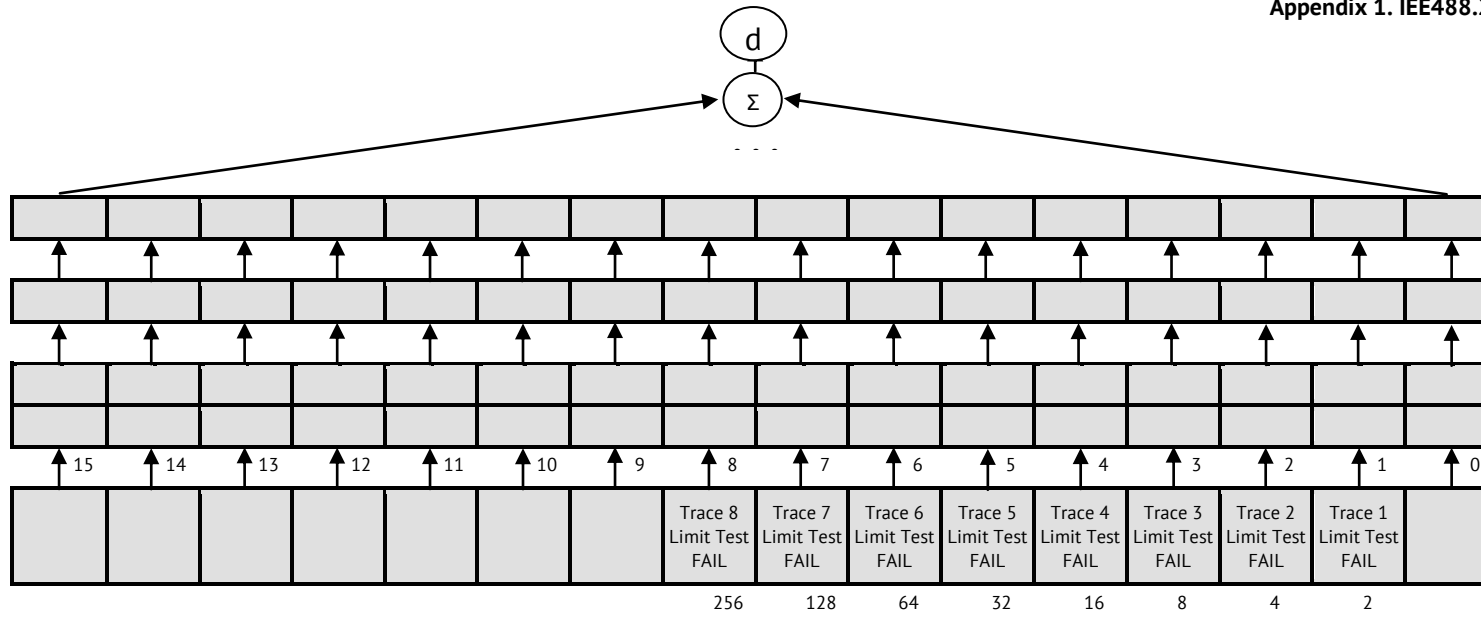
Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



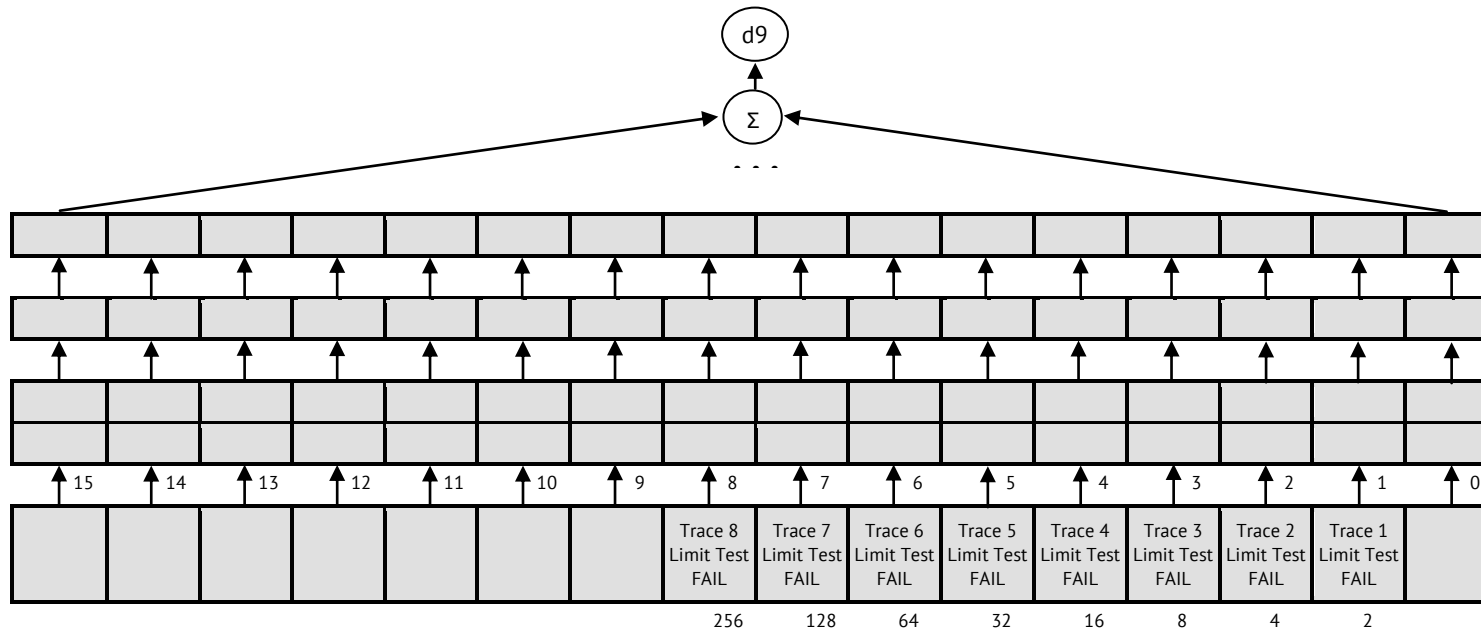
Questionable Limit Channel 1 Status Enable Register
STAT:QUES:LIM:CHAN1:ENAB

Questionable Limit Channel 1 Status Event Register
STAT:QUES:CHAN1:IM?

Positive Transition Filter
STAT:QUES:LIM:CHAN1:PTR

Negative Transition Filter
STAT:QUES:LIM:CHAN1:NTR

Questionable Limit Channel 1 Status Condition Register
STAT:QUES:LIM:CHAN1:COND?



Questionable Limit Channel 14 Status Enable Register
STAT:QUES:LIM:CHAN14:ENAB

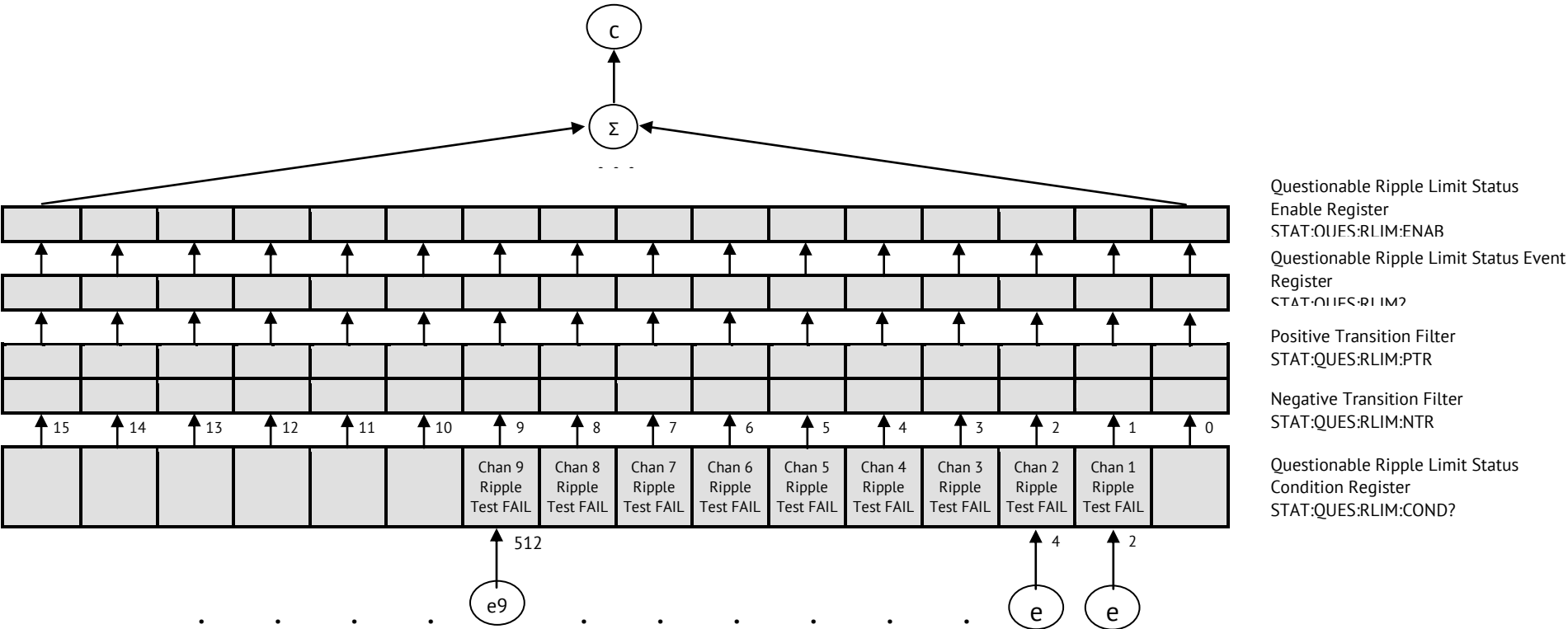
Questionable Limit Channel 9 Status Event Register
STAT:QUES:CHAN9:IM?

Positive Transition Filter
STAT:QUES:LIM:CHAN9:PTR

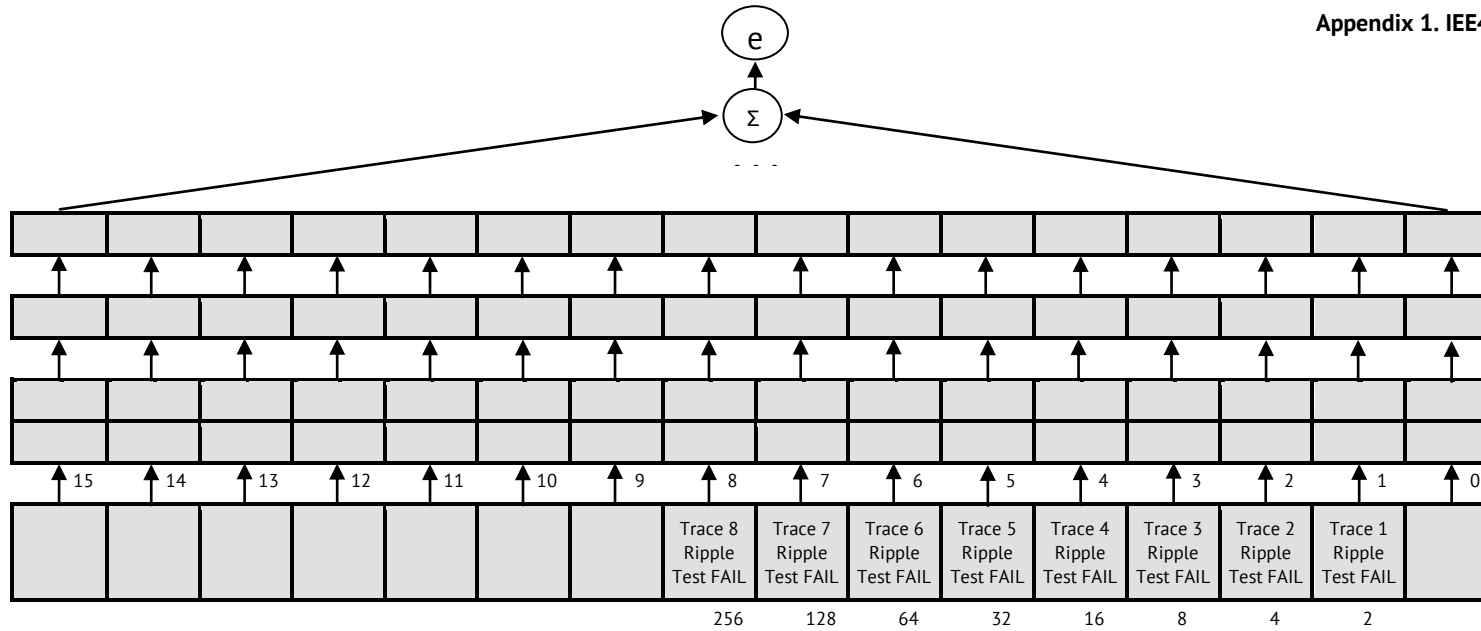
Negative Transition Filter
STAT:QUES:LIM:CHAN9:NTR

Questionable Limit Channel 9 Status Condition Register
STAT:QUES:LIM:CHAN9:COND?

Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



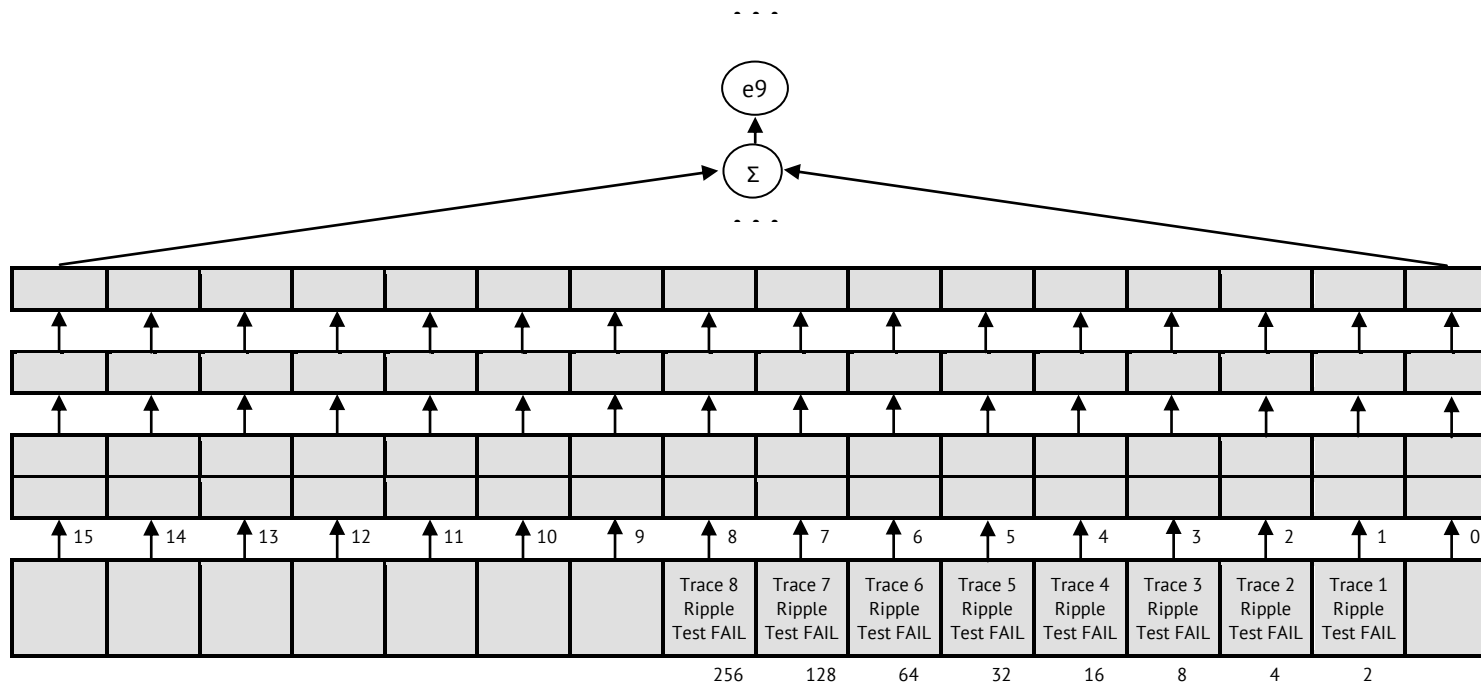
Questionable Ripple Limit Channel 1
Status Enable Register
STAT:QUES:RLIM:CHAN1:ENAB

Questionable Ripple Limit Channel 1
Status Event Register
STAT:QUES:CHAN1:RLIM?

Positive Transition Filter
STAT:QUES:RLIM:CHAN1:PTR

Negative Transition Filter
STAT:QUES:RLIM:CHAN1:NTR

Questionable Ripple Limit Channel 1
Status Condition Register
STAT:QUES:RLIM:CHAN1:COND?



Questionable Ripple Limit Channel 9
Status Enable Register
STAT:QUES:RLIM:CHAN9:ENAB

Questionable Ripple Limit Channel 9
Status Event Register
STAT:QUES:CHAN9:RLIM?

Positive Transition Filter
STAT:QUES:RLIM:CHAN9:PTR

Negative Transition Filter
STAT:QUES:RLIM:CHAN9:NTR

Questionable Ripple Limit Channel 9
Status Condition Register
STAT:QUES:RLIM:CHAN9:COND?

Appendix 2. Error Codes

114	"Header suffix out of range"
200	"Execution error"
211	"Trigger ignored"
213	"Init ignored"
220	"Parameter Error"
222	"Data out of range"
224	"Illegal parameter value"
201	"Invalid channel index"
202	"Invalid trace index"
203	"Invalid marker index"
204	"Marker is not active"
205	"Invalid save type specifier"
206	"Invalid sweep type specifier"
207	"Invalid trigger source specifier"
208	"Invalid measurement parameter specifier"
209	"Invalid format specifier"
210	"Invalid data math specifier"
214	"Invalid limit data"
215	"Invalid segment data"
216	"Invalid standard type specifier"
217	"Invalid conversion specifier"
218	"Invalid gating shape specifier"
219	"Invalid gating type specifier"
300	"Device-specific error"
302	"Status reporting system error"

Example 1. Program Written in C

The following program shows the control over the instrument using the C language with the VISA library.

The instrument address is assigned in VISA Resource Name format in the command line at the start of the program. For more detail on VISA Resource Name see the VISA library documentation.

Program description:

1. Sets up the communication with the instrument.
2. Reads out and displays the instrument information string.
3. Sets some parameters of the instrument.
4. Triggers the measurement and waits for the sweep completion.
5. Reads out the measurement data and the frequency values at the measurement points.
6. Displays the measurement data

```
// Example1.cpp
//
// VISA Header: visa.h (must be included)
// VISA Library: visa32.lib (must be linked with)
#include "stdafx.h"
#include "visa.h"

int main(int argc, char* argv[])
{
    ViStatus status;
    ViSession defaultRM, instr;
    ViUInt32 retCount;
    ViByte buffer[255];
    int temp;
    int NOP = 21;
    points
    const int maxCnt = 100;
    double Data[maxCnt*2];
    double Freq[maxCnt];

    if (argc < 2)
    {
        printf("\nUsage: Example4 <VISA address>\n\n");
        printf("VISA address examples:\n");
        printf("          TCPIP::nnn.nnn.nnn.nnn::INSTR\n");
        printf("          TCPIP::devicename::INSTR\n");
        return -1;
    }

    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        printf("Can't initialize VISA\n");
    }
    // Error checking
    // Communication channels
    // Return count from string I/O
    // Buffer for string I/O
    // Number of measurement
    // Maximum reading count
    // Measurement data array
    // Frequency array
}
```



```

    return -1;
}

status = viOpen(defaultRM, argv[1], VI_NULL, VI_NULL, &instr);
if (status < VI_SUCCESS)
{
    printf("Can't open VISA address: %s\n", argv[1]);
    return -1;
}
//
// Set the timeout for message-based communication
//
viSetAttribute(instr, VI_ATTR_TMO_VALUE, 5000);
//
// Read ID string from Analyzer
//
viPrintf(instr, "*IDN?\n");
viRead(instr, buffer, sizeof(buffer), &retCount);
printf("*IDN? Returned %d bytes: %.*s\n\n", retCount, retCount,
buffer);
//
// Set up the Analyzer
//
viPrintf(instr, "SYST:PRES\n");
viPrintf(instr, "SENS:SWE:POIN %d\n", NOP);
viPrintf(instr, "CALC:PAR1:DEF S21\n");
viPrintf(instr, "CALC:PAR1:SEL\n");
viPrintf(instr, "CALC:FORM MLOG\n");
viPrintf(instr, "SENS:BAND 10\n");
//
// Trigger measurement and wait for completion
//
viPrintf(instr, ":TRIG:SOUR BUS\n");
viPrintf(instr, ":TRIG:SING\n");
viQueryf(instr, "*OPC?\n", "%d", &temp);
//
// Read out measurement data
//
retCount = maxCnt * 2;
viQueryf(instr, "CALC:DATA:FDAT?\n", "%,#lf", &retCount, Data);
retCount = maxCnt;
viQueryf(instr, "SENS:FREQ:DATA?\n", "%,#lf", &retCount, Freq);
//
// Display measurement data
//
printf("%20s %20s %20s\n", "Frequency", "Data1", "Data2");
for (int i = 0; i < NOP; i++)
{
    printf("%20f %20f %20f\n", Freq[i], Data[i*2], Data[i*2+1]);
}

status = viClose(instr);
status = viClose(defaultRM);
return 0;
}

```

Example 2. Program Written in LabView

The following program shows the control over the instrument using the LabView language with the VISA library.

Below see the block diagram of the program and front panel of the program with the program execution result.

The front panel contains the entry field for the instrument name “VISA Resource Name”. For more detail on VISA Resource Name see the VISA library documentation.

The user must enter the instrument address, select the trace format in the “Format” field, and click the “Run” button. As the result of the program, the instrument information string will be displayed and the measurement trace will be plotted.

Program description:

1. Sets up the communication with the instrument.
2. Reads out and displays the instrument information string.
3. Sets some parameters of the instrument.
4. Generates the trigger and waits for the sweep completion.
5. Sets the trace format to the format entered by the user in the “Format” field.
6. Reads out the measurement data.
7. Displays the measurement data.

