



COPPER MOUNTAIN  
TECHNOLOGIES

## 1-Port VNA Series

R54

R140

R60

R180/RP180

Programming Manual SCPI



Software version 19.3.2  
August 2019

## TABLE OF CONTENTS

<b>1 Introduction .....</b>	<b>11</b>
1.1 Programming Manual Scope .....	11
1.2 Related Documents.....	11
1.3 References.....	11
<b>2 Analyzer Interfaces .....</b>	<b>12</b>
2.1 Ethernet Interface .....	12
<b>3 SCPI Overview.....</b>	<b>13</b>
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 .....	17
3.6.2 Booleans.....	17
3.6.3 Character Data .....	17
3.6.4 String Parameters .....	18
3.6.5 Numeric Lists.....	18
3.7 Query Commands .....	19
3.8 Numeric Suffixes.....	19
3.9 Compound Commands .....	19
3.10 IEEE488.2 Common Commands Overview.....	20
3.11 String Terminator.....	20
<b>4 Remote Control Commands .....</b>	<b>21</b>
4.1 Conventions.....	21
4.1.1 Syntax.....	21
4.1.2 Identifiers.....	22
4.2 IEEE488.2 Common Commands.....	23
*CLS .....	23
*ESE .....	24
*ESR? .....	24
*IDN? .....	25
*OPC .....	25
*OPC? .....	26
*RST .....	26
*SRE .....	27
*STB? .....	27
*TRG .....	28

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

CALC:LIM:REP:POIN?	67
CALC:LIM:REP?	68
CALC:MARK	69
CALC:MARK:ACT	69
CALC:MARK:BWID	70
CALC:MARK:BWID:DATA?	70
CALC:MARK:BWID:REF	71
CALC:MARK:BWID:THR	72
CALC:MARK:BWID:TYPE	73
CALC:MARK:COUN	73
CALC:MARK:COUP	74
CALC:MARK:FUNC:DOM	74
CALC:MARK:FUNC:DOM:STAR	75
CALC:MARK:FUNC:DOM:STOP	76
CALC:MARK:FUNC:EXEC	77
CALC:MARK:FUNC:PEXC	78
CALC:MARK:FUNC:PPOL	79
CALC:MARK:FUNC:TARG	80
CALC:MARK:FUNC:TRAC	81
CALC:MARK:FUNC:TTR	82
CALC:MARK:FUNC:TYPE	83
CALC:MARK:REF	84
CALC:MARK:SET	84
CALC:MARK:X	85
CALC:MARK:Y?	86
CALC:MATH:DEL	86
CALC:MATH:FUNC	87
CALC:MATH:MEM	88
CALC:MST	88
CALC:MST:DATA?	89
CALC:MST:DOM	89
CALC:MST:DOM:STAR	90
CALC:MST:DOM:STOP	91
CALC:PAR:COUN	92
CALC:PAR:DEF	93
CALC:PAR:SEL	94
CALC:RLIM	94
CALC:RLIM:DATA	95
CALC:RLIM:DISP:LINE	96
CALC:RLIM:FAIL?	96
CALC:RLIM:REP?	97
CALC:SMO	97
CALC:SMO:APER	98
CALC:TRAC:DATA:FDAT?	99

CALC:TRAC:DATA:FMEM?	100
CALC:TRAC:DATA:SDAT?	101
CALC:TRAC:DATA:SMEM?	102
CALC:TRAN:TIME	103
CALC:TRAN:TIME:CENT	104
CALC:TRAN:TIME:IMP:WIDT	105
CALC:TRAN:TIME:KBES	106
CALC:TRAN:TIME:LPFR	106
CALC:TRAN:TIME:SPAN	107
CALC:TRAN:TIME:STAR	108
CALC:TRAN:TIME:STAT	109
CALC:TRAN:TIME:STEP:RTIM	110
CALC:TRAN:TIME:STIM	111
CALC:TRAN:TIME:STOP	112
CALC:TRAN:TIME:UNIT	113
DEV:ADDN	114
DEV:ADDN	114
DEV:COUN?	114
DEV:MOVD	114
DEV:MOVU	115
DEV:READ?	115
DEV:ADJ:EXEC	115
DEV:ADJ:PER	116
DEV:PORT#:OFF	116
DEV:REML	117
DEV:SER?	117
DEV:METH?	118
DISP:COL:BACK	119
DISP:COL:GRAT	119
DISP:COL:RES	120
DISP:COL:TRAC:DATA	120
DISP:COL:TRAC:MEM	121
DISP:ENAB	121
DISP:FSIG	122
DISP:IMAG	123
SYST:HIDE	123
DISP:MAX	124
SYSTem:SHOW	124
DISP:SPL	125
DISP:WIND:ACT	126
DISP:WIND:ANN:MARK:ALIG	126
DISP:WIND:ANN:MARK:SING	127
DISP:WIND:MAX	127
DISP:WIND:SPL	128

DISP:WIND:TITL .....	129
DISP:WIND:TITL:DATA.....	129
DISP:WIND:TRAC:ANN:MARK:MEM.....	130
DISP:WIND:TRAC:ANN:MARK:POS:X .....	131
DISP:WIND:TRAC:ANN:MARK:POS:Y.....	132
DISP:WIND:TRAC:Y:AUTO .....	132
DISP:WIND:TRAC:Y:PDIV .....	133
DISP:WIND:TRAC:Y:RLEV .....	134
DISP:WIND:TRAC:Y:RPOS .....	135
DISP:WIND:Y:DIV .....	136
FORM:BORD .....	137
FORM:DATA.....	138
HCOP .....	139
HCOP:DATE:STAM.....	139
HCOP:IMAG.....	140
HCOP:PAIN.....	140
INIT .....	141
INIT:CONT .....	142
MMEM:COPY.....	142
MMEM:DEL.....	143
MMEM:LOAD .....	143
MMEM:LOAD:CHAN .....	144
MMEM:LOAD:CKIT.....	145
MMEM:LOAD:LIM.....	146
MMEM:LOAD:RLIM.....	146
MMEM:LOAD:SEGM .....	147
MMEM:MDIR.....	147
MMEM:STOR.....	148
MMEM:STOR:CHAN.....	148
MMEM:STOR:CHAN:CLE.....	149
MMEM:STOR:CKIT .....	149
MMEM:STOR:FDAT.....	150
MMEM:STOR:IMAG.....	150
MMEM:STOR:LIM.....	151
MMEM:STOR:RLIM .....	151
MMEM:STOR:SEGM.....	152
MMEM:STOR:SNP .....	152
MMEM:STOR:SNP:FORM.....	153
MMEM:STOR:SNP:TYPE:S1P .....	153
MMEM:STOR:SNP:TYPE:S2P .....	154
MMEM:STOR:STYP .....	154
MMEM:TRAN? .....	155
OUTP .....	155
SENS:AVER .....	156

SENS: AVER: CLE .....	156
SENS: AVER: COUN .....	157
SENS: BAND .....	157
SENS: BWID .....	158
SENS: CORR: CLE .....	158
SENS: CORR: COEF .....	159
SENS: CORR: COEF: METH: OPEN .....	160
SENS: CORR: COEF: METH: SHOR .....	161
SENS: CORR: COEF: METH: SOLT1 .....	162
SENS: CORR: COEF: SAVE .....	163
SENS: CORR: COLL: CKIT .....	164
SENS: CORR: COLL: CKIT: LAB .....	165
SENS: CORR: COLL: CKIT: RES .....	166
SENS: CORR: COLL: CKIT: STAN: C0 .....	166
SENS: CORR: COLL: CKIT: STAN: C1 .....	167
SENS: CORR: COLL: CKIT: STAN: C2 .....	168
SENS: CORR: COLL: CKIT: STAN: C3 .....	169
SENS: CORR: COLL: CKIT: STAN: DEL .....	170
SENS: CORR: COLL: CKIT: STAN: L0 .....	171
SENS: CORR: COLL: CKIT: STAN: L1 .....	172
SENS: CORR: COLL: CKIT: STAN: L2 .....	173
SENS: CORR: COLL: CKIT: STAN: L3 .....	174
SENS: CORR: COLL: CKIT: STAN: LAB? .....	175
SENS: CORR: COLL: CKIT: STAN: LOSS .....	176
SENS: CORR: COLL: CKIT: STAN: TYPE? .....	177
SENS: CORR: COLL: CKIT: STAN: Z0 .....	178
SENS: CORR: COLL: CLE .....	178
SENS: CORR: COLL: ECAL: CHECK: EXEC .....	179
SENS: CORR: COLL: ECAL: ORI: EXEC .....	179
SENS: CORR: COLL: ECAL: ORI: STAT .....	179
SENS: CORR: COLL: ECAL: SOLT1 .....	180
SENS: CORR: COLL: ECAL: SOLT2 .....	180
SENS: CORR: COLL: ECAL: UCH .....	181
SENS: CORR: COLL: LOAD .....	182
SENS: CORR: COLL: METH: ERES .....	183
SENS: CORR: COLL: METH: OPEN .....	184
SENS: CORR: COLL: METH: SHOR .....	185
SENS: CORR: COLL: METH: SOLT1 .....	186
SENS: CORR: COLL: METH: THRU .....	187
SENS: CORR: COLL: METH: TYPE? .....	188
SENS: CORR: COLL: OPEN .....	188
SENS: CORR: COLL: SAVE .....	189
SENS: CORR: COLL: SHOR .....	190
SENS: CORR: COLL: THRU .....	191

SENS:CORR:EXT .....	192
SENS:CORR:EXT:PORT:FREQ .....	193
SENS:CORR:EXT:PORT:INCL.....	194
SENS:CORR:EXT:PORT:LDC .....	194
SENS:CORR:EXT:PORT:LOSS.....	195
SENS:CORR:EXT:PORT:TIME .....	196
SENS:CORR:IMP .....	197
SENS:CORR:STAT.....	198
SENS:CORR:TRAN:TIME:FREQ.....	199
SENS:CORR:TRAN:TIME:LOSS.....	200
SENS:CORR:TRAN:TIME:RVEL.....	201
SENS:CORR:TRAN:TIME:STAT.....	202
SENS:CORR:TYPE?.....	202
SENS:FREQ:DATA? .....	203
SENS:FREQ:CENT.....	204
SENS:FREQ:SPAN.....	205
SENS:FREQ:STAR.....	206
SENS:FREQ: STOP .....	207
SENS:ROSC:SOUR.....	208
SENS:SEGM:DATA.....	209
SENS:SWE:POIN .....	210
SENS:SWE:POIN:TIME .....	211
SENS:SWE:TYPE .....	212
SERV:CHAN:ACT? .....	212
SERV:CHAN:COUN?.....	212
SERV:CHAN:TRAC:ACT?.....	213
SERV:CHAN:TRAC:COUN?.....	213
SERV:PORT:COUN? .....	213
SERV:SWE:FREQ:MAX? .....	214
SERV:SWE:FREQ:MIN? .....	214
SERV:SWE:POIN? .....	215
SOUR:POW .....	216
SOUR:POW:STAT .....	217
STAT:OPER?.....	218
STAT:OPER:COND? .....	218
STAT:OPER:ENAB .....	219
STAT:OPER:NTR .....	220
STAT:OPER:PTR.....	221
STAT:PRES .....	221
STAT:QUES:COND? .....	222
STAT:QUES:ENAB .....	222
STAT:QUES:LIM:CHAN:COND?.....	223
STAT:QUES:LIM:CHAN:ENAB.....	223
STAT:QUES:LIM:CHAN:NTR .....	224



STAT:QUES:LIM:CHAN:PTR.....	225
STAT:QUES:LIM:CHAN?.....	225
STAT:QUES:LIM:COND?.....	226
STAT:QUES:LIM:ENAB.....	226
STAT:QUES:LIM:NTR.....	227
STAT:QUES:LIM:PTR.....	228
STAT:QUES:LIM?.....	228
STAT:QUES:NTR.....	229
STAT:QUES:PTR.....	230
STAT:QUES:RLIM:CHAN:COND?.....	230
STAT:QUES:RLIM:CHAN:ENAB.....	231
STAT:QUES:RLIM:CHAN:NTR.....	232
STAT:QUES:RLIM:CHAN:PTR.....	233
STAT:QUES:RLIM:CHAN?.....	233
STAT:QUES:RLIM:COND?.....	234
STAT:QUES:RLIM:ENAB.....	234
STAT:QUES:RLIM:NTR.....	235
STAT:QUES:RLIM:PTR.....	236
STAT:QUES:RLIM?.....	236
STAT:QUES?.....	237
SYST:COMM:ECAL:TEMP:SENS?.....	237
SYST:CORR.....	238
SYST:DATE.....	238
SYST:ERR?.....	239
SYST:LOC.....	239
SYST:PRES.....	240
SYST: READY?.....	240
SYST:REM.....	241
SYST:RWL.....	241
SYST:TEMP:SENS?.....	242
SYST:TIME.....	243
TRIG.....	243
TRIG:EXT:DEL.....	244
TRIG:EXT:POS.....	245
TRIG:EXT:SLOP.....	246
TRIG:OUTP:FUNC.....	247
TRIG:OUTP:POL.....	248
TRIG:OUTP:STAT.....	249
TRIG:POIN.....	250
TRIG:SING.....	251
TRIG:SOUR.....	252
TRIG:WAIT.....	253
<b>Appendix 1. IEE488.2 Status Reporting System.....</b>	<b>254</b>
<b>Appendix 2. Error Codes.....</b>	<b>260</b>

**Appendix 3. Programming Examples ..... 261**

## 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 R54, R140, R60 and R180 models VNA.

### 1.2 Related Documents

Before reading this Manual, familiarize yourself with VNA 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

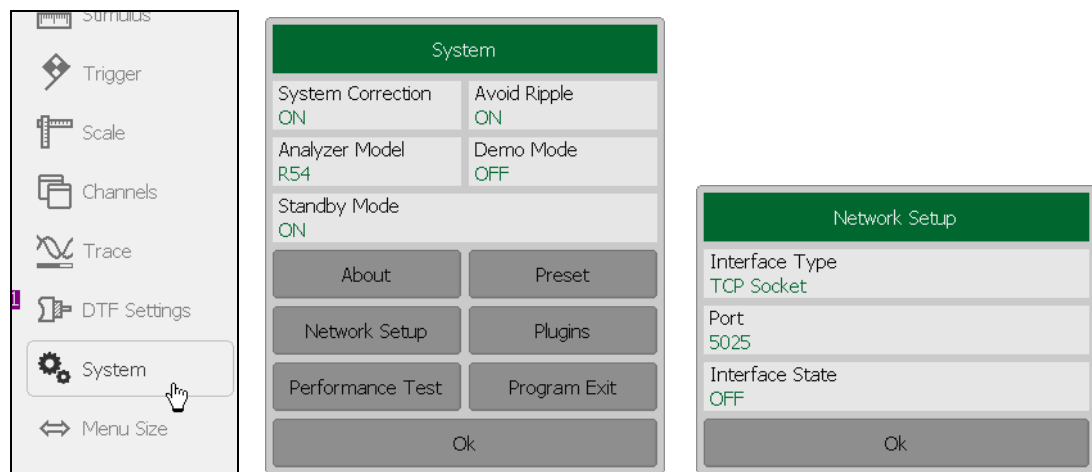
### 2.1 Ethernet Interface

Ethernet interface is part of the equipment of a personal computer that connects to the device “R54”, “R140”, “R60” or “R180”.

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 Socket protocol use the following softkeys:  
**System > Network Setup > Interface State > ON/OFF.**

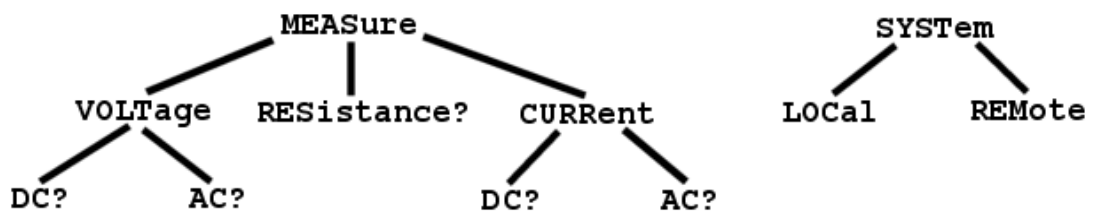
### 3 SCPI Overview

The Ethernet interface implement the 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 the 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.

<b>Prefix</b>	<b>Multiplier</b>
<b>A</b>	1e-18
<b>F</b>	1e-15
<b>P</b>	1e-12
<b>N</b>	1e-9
<b>U</b>	1e-6
<b>M</b>	1e-3
<b>K</b>	1e3
<b>MA</b>	1e6
<b>G</b>	1e9
<b>T</b>	1e12
<b>PE</b>	1e15
<b>EX</b>	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>#B</b>	#B11001010 = 202 <sub>10</sub>
Octal	<b>#Q</b>	#Q107 = 71 <sub>10</sub>
Hexadecimal	<b>#H</b>	#H10FF = 4351 <sub>10</sub>

### 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 asterix (\*). 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 indicate 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
...	ellipsis 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"> <li>• Error Queue</li> <li>• Status Byte Register</li> <li>• Standard Event Status Register</li> <li>• Operation Status Event Register</li> <li>• Questionable Status Event Register</li> <li>• Questionable Limit Status Event Register</li> <li>• Questionable Limit Channel Status Event Register</li> </ul>
<i>Target</i>	Status Reporting System
<i>Equivalent Softkeys</i>	<b>None</b>

**\*ESE**

\*ESE &lt;numeric&gt;

\*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>	<b>None</b>

**\*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>	<b>None</b>



**\*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>	<b>None</b>

**\*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>	<b>None</b>

**\*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>	<b>None</b>

**\*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 Hold.
<i>Target</i>	Instrument
<i>Related Commands</i>	SYSTem:PRESet
<i>Equivalent Softkeys</i>	<b>None</b>

**\*SRE**

\*SRE &lt;numeric&gt;

\*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>	<b>None</b>

**\*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>	<b>None</b>

**\*TRG****\*TRG**

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

**\*WAI****\*WAI**

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

### 4.3 Vector 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>	<b>None</b>

#### 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 }
<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>	<b>Analysis &gt; Conversion &gt; Conversion</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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. Error code: 217
<i>Query Response</i>	{ IMP   ADM   INV   CONJ }
<i>Preset Value</i>	IMP
<i>Equivalent Softkeys</i>	<b>Analysis &gt; Conversion &gt; Function { Impedance Z   Admittance Y   Inverse 1/S   Conjugation }</b>

## 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 }
<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>	<b>Scale &gt; Electrical Delay</b>

## 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 }
<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>	<b>Scale &gt; Phase Offset</b>



## CALC:DATA:FDAT?

CALCulate&lt;Ch&gt;[: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>&lt;numeric 2n-1&gt;    real number in rectangular format, real part in polar and Smith chart formats;</p> <p>&lt;numeric 2n&gt;        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 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	<b>None</b>

## CALC:DATA:FMEM?

CALCulate&lt;Ch&gt;[: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.</p> <p>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>&lt;numeric 2n-1&gt; real number in rectangular format, real part in polar and Smith chart formats;</p> <p>&lt;numeric 2n&gt; 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 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	<b>None</b>

## CALC:DATA:SDAT?

CALCulate&lt;Ch&gt;[: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>&lt;numeric 2n-1&gt;    the real part of corrected measurement;</p> <p>&lt;numeric 2n&gt;        the imaginary part of corrected measurement</p>
<i>Target</i>	<p>The active trace of channel &lt;Ch&gt;, &lt;Ch&gt;={ [1]   2   3   4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	<b>None</b>

## CALC:DATA:SMEM?

CALCulate&lt;Ch&gt;[: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>&lt;numeric 2n-1&gt;    the real part of corrected measurement memory;</p> <p>&lt;numeric 2n&gt;     the imaginary part of corrected measurement memory.</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	<b>None</b>

## 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 }
<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. Error code: 219
<i>Query Response</i>	{BPAS   NOTC}
<i>Preset Value</i>	BPAS
<i>Equivalent Softkeys</i>	<b>Analysis &gt; Gating &gt; Type</b>

## CALC:FILT:TIME:CENr

```
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 }
<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>	<b>None</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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. Error code: 218
<i>Query Response</i>	{MAX   WIDE   NORM   MIN }
<i>Preset Value</i>	NORM
<i>Equivalent Softkeys</i>	<b>Analysis &gt; Gating &gt; Shape &gt;{ Maximum   Wide   Normal   Minimum }</b>

## 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 }
<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>	<b>None</b>



## 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 }
<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>	<b>Analysis &gt; Gating &gt; Start</b>

## 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 }
<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>	<b>Analysis &gt; Gating &gt; Gating</b>

## 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 }
<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>	<b>Analysis &gt; Gating &gt; Stop</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; specifies the trace format:</p> <p>MLOGarithmic : Logarithmic magnitude</p> <p>PHASe : Phase</p> <p>UPHase : Expanded phase</p> <p>GDELay : Group delay time</p> <p>SWR : Voltage standing wave ratio</p> <p>REAL : Real part</p> <p>IMAGinary : Imaginary part</p> <p>MLINear : Linear magnitude</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>CLOS : Cable Loss - Logarithmic magnitude</p> <p>RLOSs : Retutn Loss (obsolete, write only) use MLOG<sup>1</sup></p> <p>DSWR : DFT in SWR chart format (obsolete, write only) use SWR+Time Domain<sup>1</sup></p> <p>DRLOSs : DFT Logarithmic magnitude (obsolete, write only) useMLOG+Time Domain<sup>1</sup></p>
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{MLOG   PHAS   UPH   GDEL   SWR   REAL   IMAG   MLIN   SLIN   SLOG   SCOM   SMITH   SADM   PLIN   PLOG   POL   CLOS }
<i>Preset Value</i>	MLOG
<i>Equivalent Softkeys</i>	<b>Trace &gt; Format&gt; Log Magnitude   Phase   Expand Phase   Group Delay   SWR   Real   Imag   Lin Magnitude   Smith (Lin)   Smith (Log)   Smith (Re/Im)   Smith (R+jX)   Smith (G+jB)   Polar (Lin)   Polar (Log)   Polar (Re/Im) Cable Loss</b>

<sup>1</sup>To switch on Time Domain mode – Analysis > Time Domain > Time Domain

## 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 } <Pt>={ [ 1 ] }
<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>	<b>Analysis &gt; Fixture Simulator &gt; De-Embedding</b>

## 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 } <Pt>={ [1] }
<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>	<b>Analysis &gt; Fixture Simulator &gt; De-Embedding &gt; S-parameters File</b>

## 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 } <Pt>={ [ 1 ] }
<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>	<b>Analysis &gt; Fixture Simulator &gt; Embedding</b>

## CALC:FSIM:SEND:PMC:PORT:USER:FILE

```
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 } <Pt>={ [1] }
<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>	<b>Analysis &gt; Fixture Simulator &gt; S-parameters File (Embedding)</b>



## 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 } <Pt>={ [ 1 ] }
<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>	<b>Analysis &gt; Fixture Simulator &gt; Port Z0</b>

## 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 }
<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>	<b>Analysis &gt; Fixture Simulator &gt; Port Z Conversion</b>

## CALC:FUNC:DATA?

```
CALCulate<Ch>[:SELEcted]:FUNctIon:DATA?
```

<i>Description</i>	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: <numeric 2n-1> the response value in n-th measurement point; <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.
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric 2N> The data transfer format depends on the FORM:DATA command setting.
<i>Related Commands</i>	CALC:FUNC:EXEC CALC:FUNC:POIN? FORM:DATA
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>None</b>

## CALC:FUNC:EXEC

CALCulate&lt;Ch&gt;[: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 }
<i>Related Commands</i>	CALC:FUNC:TYPE CALC:FUNC:DATA?
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>None</b>

## CALC:FUNC:POIN?

CALCulate&lt;Ch&gt;[:SElected]:FUNction:POINts?

<i>Description</i>	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.
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<numeric>
<i>Related Commands</i>	CALC:FUNC:EXEC
<i>Equivalent Softkeys</i>	<b>None</b>



## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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>	<b>None</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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>	<b>None</b>

## 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 }
<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>	<b>Analysis &gt; Limit Test &gt; Limit Test</b>

## 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 <math>1 + 5N</math>, where <math>N</math> is the number of limit line segments. For the <math>n</math>-th point, where <math>n</math> from 1 to <math>N</math>:</p> <p>&lt;numeric 1&gt;            the number of limit line segments <math>N</math> is from 0 to 100. Setting 0 clears the limit line.</p> <p>&lt;numeric 5n -3&gt;    type of the <math>n</math>-th limit line segment 0: Off 1: Upper limit 2: Lower limit 3: Single Point limit</p> <p>&lt;numeric 5n-2&gt;    the stimulus value in the start point of the <math>n</math>-th segment</p> <p>&lt;numeric 5n-1&gt;    the stimulus value in the end point of the <math>n</math>-th segment</p> <p>&lt;numeric 5n-0&gt;    the response value in the start point of the <math>n</math>-th segment</p> <p>&lt;numeric 5n+1&gt;    the response value in the end point of the <math>n</math>-th segment</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 5N+1&gt;</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 <numeric 1>, an error occurs (error code 214). If <numeric 5n-3> is less than 0 or more than 2, an error occurs (error code 214). When <numeric 5n-2>, <numeric 5n-1>, <numeric 5n-0>, and <numeric 5n+1> 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>	<b>Analysis &gt; Limit Test &gt; Edit Limit Line</b>

## 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 }
<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>	<b>Analysis &gt; Limit Test &gt; Limit Line</b>

## 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 }
<i>Query Response</i>	1      Fail 0      Pass
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>Analysis &gt; Limit Test &gt; Response Offset</b>



## 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 }
<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>	<b>Analysis &gt; Limit Test &gt; Stimulus Offset</b>

## CALC:LIM:REP:ALL?

CALCulate&lt;Ch&gt;[: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>&lt;numeric 4n-3&gt; the stimulus value in the n-th point;</p> <p>&lt;numeric 4n-2&gt; the limit test result in the n-th point;-1: No limit 0: Fail 1: Pass</p> <p>&lt;numeric 4n-1&gt; the upper limit value in the n-th point (0 - if there is no limit)</p> <p>&lt;numeric 4n-0&gt; 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 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 4N&gt;</p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	<b>None</b>

## CALC:LIM:REP:POIN?

CALCulate&lt;Ch&gt;[: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 }
<i>Query Response</i>	<numeric>
<i>Related Commands</i>	CALC:LIM:REP?
<i>Equivalent Softkeys</i>	<b>None</b>

## CALC:LIM:REP?

CALCulate&lt;Ch&gt;[: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 }
<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>	<b>None</b>

## 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 } <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>	<b>Marker &gt; Add Marker   Remove Marker</b> <b>Marker &gt; Reference Marker</b>

## CALC:MARK:ACT

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

<i>Description</i>	Sets the active marker (no query).  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.
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1]  2 3 4 } <Mk>={ [1]  2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 }
<i>Equivalent Softkeys</i>	<b>None</b>

## CALC:MARK:BWID

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

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

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

## CALC:MARK:BWID:DATA?

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

<i>Description</i>	Reads out the bandwidth search result (query only). The bandwidth search can be 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: <numeric 1> Bandwidth; <numeric 2> Center frequency; <numeric 3> Q value; <numeric 4> Loss;
<i>Target</i>	Marker <Mk> of the active trace of channel <Ch>, <Ch>={ [1]   2   3   4 } <Mk>={ [1]   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16 }
<i>Query Response</i>	<numeric 1>, <numeric 2>, ...<numeric 4>
<i>Related Commands</i>	CALC:MARK:BWID:REF
<i>Notes</i>	If the bandwidth search is impossible, all the read out values are 0. If the search is performed relatively to a marker, which is OFF, an error occurs (error code 204).
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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>	<b>Marker &gt; Math &gt; Bandwidth Search &gt; Search Ref To</b>

## 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 } <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>	<b>Marker &gt; Math &gt; Bandwidth Search &gt; Bandwidth Value</b>



## 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 }
<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>	<b>Marker &gt; Math &gt; Bandwidth Search &gt; Type Search</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>Marker &gt; Properties &gt; Marker Couple</b>

## CALC:MARK:FUNC:DOM

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

```
CALCulate<Ch>[:SElected]:MARKer:FUNCtion: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 }
<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>	<b>Marker &gt; Search &gt; Search Range</b>

## 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 }
<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>Equivalent Softkeys</i>	<b>Markers &gt; Search &gt; Search Start</b>

## 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 }
<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>Equivalent Softkeys</i>	<b>Markers &gt; Search &gt; Search Stop</b>

## CALC:MARK:FUNC:EXEC

CALCulate&lt;Ch&gt;[:SElected]:MARKer&lt;Mk&gt;: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 } <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>	<b>Marker &gt; Search &gt; { Maximum   Minimum }</b>  <b>Marker &gt; Search &gt; Search Peak &gt; { Search Peak   Search Max Peak   Search Peak Left   Search Peak Right }</b>  <b>Marker &gt; Search &gt; Search Target &gt; { Search Target   Search Target Left   Search Target Right }</b>

## 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 } <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>	<b>Marker &gt; Search &gt; Search Peak &gt; Peak Excursion</b>

## 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 } <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>	<b>Marker &gt; Search &gt; Search Peak &gt; Peak Polarity &gt; { Positive   Negative   Both }</b>

## 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 } <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>	<b>Marker &gt; Search &gt; Search Target &gt; Target Value</b>



## 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 } <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>	<b>Marker &gt; Search &gt; Tracking</b>

## 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 } <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>	<b>Marker &gt; Search &gt; Search Target &gt; Target Transition &gt; { Positive   Negative   Both }</b>

## 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 } <Mk>={ [1]   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16 }
<i>Parameter</i>	<p>&lt;char&gt; specifies the type of the marker search:</p> <p>MAXimum : Maximum value search</p> <p>MINimum : Minimum value search</p> <p>PEAK : Peak search</p> <p>LPEak : Peak search to the left from the marker</p> <p>RPEak : Peak search to the right from the marker</p> <p>TARGet : Target search</p> <p>LTARget : Target search to the left from the marker</p> <p>RTARget : Target search to the right from the marker</p>
<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>	<p><b>Marker &gt; Search &gt; { Maximum   Minimum }</b></p> <p><b>Marker &gt; Search &gt; Peak &gt; { Search Peak   Search Max Peak   Search Peak Left   Search Peak Right }</b></p> <p><b>Marker &gt; Search &gt; Target &gt; { Search Target   Search Target Left   Search Target Right }</b></p>

## 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 }
<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>	<b>Marker &gt; Reference Marker</b>

## 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 } <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 START        : Sweep start value set to the stimulus value of the marker position. STOP         : Sweep stop value set to the stimulus value of the marker position. CENTER       : Sweep center value set to the stimulus value of the marker position. RLEVEL      : Reference value set to the response value of the marker position.
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 } <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>	<b>None</b>

## CALC:MARK:Y?

CALCulate&lt;Ch&gt;[:SElected]:MARKer&lt;Mk&gt;: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>&lt;numeric 1&gt; real number in rectangular format, real part in polar and Smith chart formats;</p> <p>&lt;numeric 2&gt; 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>Marker &lt;Mk&gt; of the active trace of channel &lt;Ch&gt;, &lt;Ch&gt;={ [1]  2 3 4} &lt;Mk&gt;={ [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>	<b>None</b>

## CALC:MATH:DEL

CALCulate&lt;Ch&gt;[:SElected]:MATH:DElete

<i>Description</i>	Removes the data from the memory trace.
<i>Target</i>	<p>The active trace of channel &lt;Ch&gt;, &lt;Ch&gt;={ [1]  2 3 4}</p>
<i>Equivalent Softkeys</i>	<b>Trace &gt; Clear Memory</b>

## 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 }
<i>Parameter</i>	<p>&lt;char&gt; 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. Error code: 210.
<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>	<b>Trace &gt; Data Math &gt; { Data/Mem   Data*Mem   Data+Mem   Data–Mem   OFF }</b>

## 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 }
<i>Equivalent Softkeys</i>	<b>Trace &gt; Memory Trace</b>

## 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 }
<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>	<b>Marker &gt; Math &gt; Statistics &gt; Statistics</b>



## 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>&lt;numeric 1&gt;    Mean value; &lt;numeric 2&gt;    Standard deviation; &lt;numeric 3&gt;    Peak-to-peak (difference between the maximum value and the minimum value).</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Related Commands</i>	CALC:MST:DOM
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>Marker &gt; Math &gt; Statistics &gt; Statistics Range</b>

## 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 }
<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>	<b>Marker &gt; Math &gt; Statistics &gt; Statistics Start</b>

## 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 }
<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>	<b>Marker &gt; Math &gt; Statistics &gt; Statistics Stop</b>

## 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 }
<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>	<b>None</b>

## CALC:PAR:DEF

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

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

<i>Description</i>	Sets the measurement parameter of the trace (command/query) (in multiport mode only).
<i>Target</i>	Trace <Tr> of channel <Ch>, <Tr>={ [1]   2   3   4 } <Ch>={ [1]   2   3   4 }  <Tr>={ [1] ...   16 } (in multiport mode only). <Ch>={ [1] ...   16 } (in multiport mode only).
<i>Parameter</i>	{ S11 S21 ... S16.1 ... S1.16 ...  S16.16 }
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Query Response</i>	{ S11   S21   ...   S16.1   ...   S16.16 }
<i>Equivalent Softkeys</i>	<b>Trace &gt; Measurement</b> (in multiport mode only)

## 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 } <Ch>={ [ 1 ]   2   3   4 }
<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>	<b>Trace &gt; Active Trace</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 }
<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>	<b>Analysis &gt; Ripple Test &gt; Ripple Test</b>

## 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 <math>1 + 4N</math>, where <math>N</math> is the number of limit line segments. For the <math>n</math>-th point, where <math>n</math> from 1 to <math>N</math>:</p> <p>&lt;numeric 1&gt;            the number of limit line segments <math>N</math> is the integer from 0 to 12. Setting 0 clears the limit line.</p> <p>&lt;numeric <math>4n - 2</math>&gt;    type of the <math>n</math>-th limit line segment 0: Off. 1: On</p> <p>&lt;numeric <math>4n - 1</math>&gt;    the stimulus value in the beginning point of the <math>n</math>-th segment</p> <p>&lt;numeric <math>4n - 0</math>&gt;    the stimulus value in the end point of the <math>n</math>-th segment</p> <p>&lt;numeric <math>4n + 1</math>&gt;    the ripple limit value of the <math>n</math>-th segment</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric <math>4N + 1</math>&gt;</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 <numeric 1>, an error occurs (error code 214). If <numeric $4n - 2$ > is less than 0 or more than 1, an error occurs (error code 214). When <numeric $4n - 1$ >, <numeric $4n - 0$ >, and <numeric $4n + 1$ > 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>	<b>Analysis &gt; Ripple Test &gt; Edit Ripple Limit</b>

## 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 }
<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>	<b>Analysis &gt; Ripple Test &gt; Limit Line</b>

## 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 }
<i>Query Response</i>	1     Fail 0     Pass
<i>Equivalent Softkeys</i>	<b>None</b>



## 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 <math>1+3N</math>, where <math>N</math> is the number of ripple limit bands. For the <math>n</math>-th point, where <math>n</math> from 1 to <math>N</math>:</p> <p>&lt;numeric 1&gt;            <math>N</math> total number of the bands          &lt;numeric <math>3n-1</math>&gt;    <math>n</math> number of the band          &lt;numeric <math>3n-0</math>&gt;    Ripple value in the <math>n</math>-th band          &lt;numeric <math>3n+1</math>&gt;    Ripple limit test result in the <math>n</math>-th band:                                    0: Pass                                    1: Fail</p>
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<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>	<b>None</b>

## 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 }
<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>	<b>Average &gt; Smoothing</b>

## 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 }
<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>	<b>Average &gt; Smoothing Aperture</b>

## CALC:TRAC:DATA:FDAT?

CALCulate&lt;Ch&gt;:TRACe&lt;Tr&gt;: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>&lt;numeric 2n-1&gt;    real number in rectangular format, real part in polar and Smith chart formats;</p> <p>&lt;numeric 2n&gt;        0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>The specified trace &lt;Tr&gt; of channel &lt;Ch&gt;, &lt;Tr&gt;={ [1]  2 3 4} &lt;Ch&gt;={ [1]  2 3 4}</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	<b>None</b>

## CALC:TRAC:DATA:FMEM?

CALCulate&lt;Ch&gt;:TRACe&lt;Tr&gt;: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.</p> <p>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>&lt;numeric 2n-1&gt; real number in rectangular format, real part in polar and Smith chart formats;</p> <p>&lt;numeric 2n&gt; 0 in rectangular format, imaginary part in polar and Smith chart formats.</p>
<i>Target</i>	<p>The specified trace &lt;Tr&gt; of channel &lt;Ch&gt;, &lt;Tr&gt;={ [1]  2 3 4 } &lt;Ch&gt;={ [1]  2 3 4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	CALC:DATA:FMEM
<i>Equivalent Softkeys</i>	<b>None</b>

## CALC:TRAC:DATA:SDAT?

CALCulate&lt;Ch&gt;:TRACe&lt;Tr&gt;: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>&lt;numeric 2n-1&gt;    the real part of corrected measurement;</p> <p>&lt;numeric 2n&gt;        the imaginary part of corrected measurement.</p>
<i>Target</i>	<p>The specified trace &lt;Tr&gt; of channel &lt;Ch&gt;,  &lt;Tr&gt;={ [1]  2 3 4 }  &lt;Ch&gt;={ [1]  2 3 4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	<b>None</b>

## CALC:TRAC:DATA:SMEM?

CALCulate&lt;Ch&gt;:TRACe&lt;Tr&gt;: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>&lt;numeric 2n-1&gt; the real part of corrected measurement memory; &lt;numeric 2n&gt; the imaginary part of corrected measurement memory.</p>
<i>Target</i>	<p>The specified trace &lt;Tr&gt; of channel &lt;Ch&gt;, &lt;Tr&gt;={ [1]  2 3 4 } &lt;Ch&gt;={ [1]  2 3 4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 2N&gt;</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>	CALC:DATA:SMEM
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Response Type &gt; { Bandpass   Lowpass Step   Lowpass Impulse }</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Center</b>



## 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 $\beta$ parameter. The impulse width setting changes the $\beta$ parameter, and setting of $\beta$ parameter changes the impulse width (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [ 1 ]   2   3   4 }
<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>	<b>None</b>

## 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 $\beta$ 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 }
<i>Parameter</i>	<numeric> $\beta$ 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>	<b>None</b>

## 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 }
<i>Equivalent Softkeys</i>	<b>Analysis &gt; Time Domain &gt; Set Frequency Low Pass</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Span</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Start</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Time Domain</b>

## 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 $\beta$ parameter. The impulse width setting changes the $\beta$ parameter, and setting of $\beta$ parameter changes the impulse width (command/query).
<i>Target</i>	The active trace of channel <Ch>, <Ch>={ [1]   2   3   4 }
<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>	<b>None</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Response Type &gt; { Bandpass   Lowpass Step   Lowpass Impulse }</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Stop</b>



## 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 } <Ch>={ [1]...   16 } (in multiport mode only).
<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>	<b>Analysis &gt; Time Domain &gt; Unit &gt; Time, ns   Metric, m   Imperial, ft</b>

## DEV:ADDN

DEVICES:ADDNext

<i>Description</i>	Adding one more device to configuration (in multiport mode only)
<i>Equivalent Softkeys</i>	<b>Devices &gt; Add Next</b>

## DEV:COUN?

DEVICES:COUNT

<i>Description</i>	Number of connected devices in the configuration (query only) (in multiport mode only)
<i>Equivalent Softkeys</i>	<b>None</b>

## DEV:MOVD

DEVICES:MOVDdown &lt;numeric&gt;

<i>Description</i>	Moving the device from the position <i>Port</i> into position <i>Port+1</i> (in multiport mode only)
<i>Parameter</i>	<numeric> Port Number {[1] 2 3 4... 16}
<i>Equivalent Softkeys</i>	<b>Devices &gt; Move Down</b>

## DEV:MOVU

DEVICES:MOVUp <numeric>

<i>Description</i>	Moving the device from the position <i>Port</i> into position <i>Port-1</i> (in multiport mode only)
<i>Parameter</i>	<numeric> Port Number {[1] 2 3 4... 16}
<i>Equivalent Softkeys</i>	<b>Devices &gt; Move Up</b>

## DEV:READ?

DEVICES:READY? <numeric>

<i>Description</i>	Readiness of the device associated with port <i>Port</i> (query only) (in multiport mode only)
<i>Parameter</i>	<numeric> Port Number {[1] 2 3 4... 16}
<i>Equivalent Softkeys</i>	<b>None</b>

## DEV:ADJ:EXEC

DEVICES[:REFERence]:ADJust:Execute

<i>Description</i>	Adjustment of the reference frequency of all devices relative to the first in the list (in multiport mode only)
<i>Equivalent Softkeys</i>	<b>Devices &gt; Adjust Immediate</b>

## DEV:ADJ:PER

```
DEVICES[:REFERENCE]:ADJUST:PERIOD <numeric>
```

<i>Description</i>	Frequency automatic adjustment period (command/query) (in multiport mode only)
<i>Parameter</i>	<numeric> Value of Frequency automatic adjustment period
<i>Unit</i>	sec
<i>Equivalent Softkeys</i>	<b>Devices &gt; Frequency Adjust Period { OFF   3   10   30   100   300 }</b>

## DEV:PORT#:OFF

```
DEVICES[:REFERENCE]:PORT<Pt>:OFFSET <numeric>
```

```
DEVICES[:REFERENCE]:PORT<Pt>:OFFSET?
```

<i>Description</i>	Reference frequency correction value (command/query) (in multiport mode only)
<i>Target</i>	Port <Pt>, <Pt>={ [1]   2   3   4... 16 }
<i>Parameter</i>	<numeric> Offset of Reference Frequency
<i>Unit</i>	Hz
<i>Equivalent Softkeys</i>	<b>None</b>

## DEV:REML

DEVICES:REMLast

<i>Description</i>	Removing one more device to configuration (in multiport mode only)
<i>Equivalent Softkeys</i>	<b>Devices &gt; Remove Last</b>

## DEV:SER?

DEVICES:SERial? &lt;numeric&gt;

<i>Description</i>	Serial number of the device associated with <i>Port</i> (query only) (in multiport mode only)
<i>Parameter</i>	<numeric> Port Number {[1] 2 3 4... 16}
<i>Equivalent Softkeys</i>	<b>None</b>

## DEV:METH?

```
DEVIces[:SYNChronization]:METHod <char>
```

```
DEVIces[:SYNChronization]:METHod?
```

<i>Description</i>	Device synchronization method (command/query) (in multiport mode only)
<i>Parameter</i>	<p>&lt;char&gt; choose from:</p> <p>FREE : Independent operation of devices. No synchronization.</p> <p>USB : Device Synchronization via USB Bus</p> <p>TRIG : Device Synchronization via Trigger Bus</p>
<i>Out of Range</i>	The command is ignored.
<i>Query Response</i>	{FREE USB TRIG}
<i>Preset Value</i>	USB
<i>Equivalent Softkeys</i>	<b>Devices &gt; Synchronization { Free Run   USB Bus   Trig Bus }</b>

## 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>	<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>	0, 0, 0
<i>Equivalent Softkeys</i>	<b>Display &gt; Interface Elements Color &gt; Background</b>

## 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>	<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>	160, 160, 164
<i>Equivalent Softkeys</i>	<b>Display &gt; Interface Elements Color &gt; Grid</b>

## DISP:COL:RES

DISPlay:COLor:RESet

<i>Description</i>	Restores the display settings to the default values (no query).
<i>Equivalent Softkeys</i>	<b>Display &gt; Preset</b>

## DISP:COL:TRAC:DATA

DISPlay:COLor:TRACe&lt;Tr&gt;:DATA &lt;numeric 1&gt;,&lt;numeric 2&gt;,&lt;numeric 3&gt;

DISPlay:COLor:TRACe&lt;Tr&gt;:DATA?

<i>Description</i>	Sets or reads out the data trace color (command/query).
<i>Target</i>	Trace <Tr>, <Tr>={ [1]   2   3   4 }
<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>	<b>Display &gt; Interface Elements Color &gt; Data Trace</b>



## 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 }
<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>	<b>Display &gt; Interface Elements Color &gt; Memory Trace</b>

## 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>	<b>Display &gt; Update</b>

## 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>	<b>Analysis &gt; Limit Test &gt; Fail Sign</b> <b>Analysis &gt; Ripple Test &gt; Fail Sign</b>

## DISP:IMAG

DISPlay:IMAGe &lt;char&gt;

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>	<b>Display &gt; Inverse Color</b>

## SYST:HIDE

SYSTem:HIDE

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

## 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>	<b>Channels &gt; Maximize Channel</b>

## SYSTem:SHOW

```
SYSTem: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>	<b>None</b>







## 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>	<b>Channels</b>

### Channel window layout on the screen

1: 	2: 	3: 
4: 	5: 	6: 

## DISP:WIND:ACT

DISPlay:WINDow&lt;Ch&gt;:ACTivate

<i>Description</i>	Sets the active channel (no query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1]   2   3   4 }
<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>	<b>Channels &gt; Active Channel</b>

## DISP:WIND:ANN:MARK:ALIG

DISPlay:WINDow&lt;Ch&gt;:ANNotation:MARKer:ALIGn[:TYPE] &lt;char&gt;

DISPlay:WINDow&lt;Ch&gt;: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 }
<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>	<b>Marker &gt; Properties &gt; Align &gt; { Vertical   Horizontal   OFF }</b>

## 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 }
<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>	<b>Marker &gt; Properties &gt; Active Only</b>

## 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>	<b>Trace &gt; Trace Allocation &gt; Maximize Trace</b>




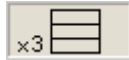


## DISP:WIND:SPL

```
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 }
<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>	<b>Trace &gt; Trace Allocation</b>

## Graph layout in the channel window

1: 	2: 	3: 
4: 	5: 	6: 



## 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 }
<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>	<b>Display &gt; Caption</b>

## 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 }
<i>Parameter</i>	<string>, up to 256 characters (quoted string)
<i>Query Response</i>	<string>
<i>Preset Value</i>	""
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>Markers &gt; Properties &gt; Memory Value</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>None</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>None</b>

## 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 } <Ch>={ [1]   2   3   4 }
<i>Equivalent Softkeys</i>	<b>Scale &gt; Auto Scale</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>Scale &gt; Scale</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>Scale &gt; Ref Value</b>

## 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 } <Ch>={ [1]   2   3   4 }
<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>	<b>Scale &gt; Ref Position</b>

## 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 }
<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>	<b>Scale &gt; Divisions</b>



## FORM:BORD

FORMat:BORDer &lt;char&gt;

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>	<b>None</b>

## FORM:DATA

FORMat:DATA &lt;char&gt;

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>&lt;char&gt; 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>	<b>None</b>

## HCOP

HCOPY[:IMMEDIATE]

<i>Description</i>	Prints out the image displayed on the screen without previewing (no query).
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>Print &gt; Print Date &amp; Time</b>

## HCOP:IMAG

HCOPy:IMAGe &lt;char&gt;

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>	<b>Print &gt; Invert Image</b>

## HCOP:PAIN

HCOPy:PAINT &lt;char&gt;

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>	<b>Print &gt; Print Color</b>

## INIT

INITiate&lt;Ch&gt;[: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 (error code 213) 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.</p> <p>(no query)</p>
<i>Target</i>	<p>Channel &lt;Ch&gt;,          &lt;Ch&gt;={ [1]   2   3   4 }</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><b>Trigger &gt;Trigger Mode &gt; Single</b></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 }
<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>	<b>Trigger &gt; Trigger Mode &gt; Continuous</b> <b>Trigger &gt; Trigger Mode &gt; Hold</b>

## 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>	<b>None</b>

## MMEM:DEL

MMEMory:DELeTe &lt;string&gt;

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

## MMEM:LOAD

MMEMory:LOAD[:STATe] &lt;string&gt;

<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>	<b>Files &gt; State &gt; Recall State</b>

## MMEM:LOAD:CHAN

MMEMory:LOAD:CHANnel[:STATe] &lt;char&gt;

<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>&lt;char&gt; choose from:</p> <p>A : Recall from register A  B : Recall from register B  C : Recall from register C  D : Recall from register D</p>
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	<b>Chanel &gt; Recall Channel &gt; { State A   B   C   D }</b>



## MMEM:LOAD:CKIT

MMEMory:LOAD:CKIT&lt;Ck&gt; &lt;string&gt;

<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   ... }
<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>	<b>None</b>

## MMEM:LOAD:LIM

MMEMory:LOAD:LIMit &lt;string&gt;

<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>	<b>Analysis &gt; Limit Test &gt; Edit Limit Line &gt; Restore Limit Table</b>

## MMEM:LOAD:RLIM

MMEMory:LOAD:RLIMit &lt;string&gt;

<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>	<b>Analysis &gt; Ripple Test &gt; Edit Ripple Limit &gt; Restore Ripple Table</b>

## MMEM:LOAD:SEGM

MMEMory:LOAD:SEGment &lt;string&gt;

<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>	<b>Stimulus &gt; Segment Table &gt; Recall...</b>

## MMEM:MDIR

MMEMory:MDIRectory &lt;string&gt;

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

## MMEM:STOR

MMEMory:STORe[:STATe] &lt;string&gt;

<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>	<b>Files &gt; State &gt; Save State...</b>

## MMEM:STOR:CHAN

MMEMory:STORe:CHANnel[:STATe] &lt;char&gt;

<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>	<char> choose from:  A : Save to register A B : Save to register B C : Save to register C D : Save to register D
<i>Out of Range</i>	The command is ignored.
<i>Equivalent Softkeys</i>	<b>Channels &gt; Save Channel &gt; { State A   B   C   D }</b>

## 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>	<b>Channels &gt; Save Channel &gt; Clear States</b>

## MMEM:STOR:CKIT

MMEMory:STORe:CKIT&lt;Ck&gt; &lt;string&gt;

<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>	<b>None</b>

## MMEM:STOR:FDAT

MMEMory:STORe:FDATa &lt;string&gt;

<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>	<b>Files &gt; Save Trace Data</b>

## MMEM:STOR:IMAG

MMEMory:STORe:IMAGe &lt;string&gt;

<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>	<b>Print &gt; Print with MS Windows</b>

## MMEM:STOR:LIM

MMEMory:STORe:LIMit &lt;string&gt;

<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>	<b>Analysis &gt; Limit Test &gt; Edit Limit Line &gt; Save Limit Table</b>

## MMEM:STOR:RLIM

MMEMory:STORe:RLIMit &lt;string&gt;

<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>	<b>Analysis &gt; Ripple Test &gt; Edit Ripple Limit &gt; Save Ripple Table</b>

## MMEM:STOR:SEGM

MMEMory:STORe:SEGMENT &lt;string&gt;

<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>	<b>Stimulus &gt; Segment Table &gt; Save...</b>

## MMEM:STOR:SNP

MMEMory:STORe:SNP[:DATA] &lt;string&gt;

<i>Description</i>	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.  (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 \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>	<b>Files &gt; Save Touchstone &gt; Save Touchstone</b>



## 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>&lt;char&gt; 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>	<b>Files &gt; Save Touchstone &gt; Touchstone Format</b>

## 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>	<b>Files &gt; Save Touchstone &gt; Type &gt; S1P</b>

## 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>	<b>Files &gt; Save Touchstone &gt; Type &gt; S2P</b>

## 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>	<char> choose from: STATe : Measurement conditions CSTate : Measurement conditions and calibration tables DSTate : Measurement conditions and data traces CDSTate : Measurement conditions, calibration tables and data traces
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Query Response</i>	{STAT CST DST CDST}
<i>Preset Value</i>	CST
<i>Equivalent Softkeys</i>	<b>Files &gt; State &gt; Save Type</b>

## MMEM:TRAN?

MMEMory:TRANsfer? &lt;string&gt;

<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>	<b>None</b>

## 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>	<b>Stimulus &gt; Power &gt; RF out Off</b>

## 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 }
<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>	<b>Average &gt; Averaging</b>

## 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 }
<i>Related Commands</i>	SENS: AVER
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>Average &gt; Averaging Factor</b>

## SENS:BAND

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

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

## SENS:BWID

SENSe&lt;Ch&gt;:BWIDth[:RESolution] &lt;frequency&gt;

SENSe&lt;Ch&gt;:BWIDth[:RESolution]?

<i>Description</i>	Sets or reads out the IF bandwidth (command/query).
<i>Target</i>	Channel <Ch>, <Ch>={ [1]   2   3   4 }
<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>	<b>Average &gt; IFBW</b>

## SENS:CORR:CLE

SENSe&lt;Ch&gt;:CORRection:CLEar

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

## SENS:CORR:COEF

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

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

<i>Description</i>	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:  <numeric 2n-1>    real part of the calibration coefficients  <numeric 2n>       imaginary part of the calibration coefficients
<i>Target</i>	Channel <Ch>, <Ch>={ [1]  2 3 4 }
<i>Parameter</i>	<char> specifies the Error term:  ER            : Reflection tracking  ED            : Directivity  ES            : Source match  <rcvport>, the number of the receiver port = 1 <srcport>, the number of the source port = 1 <numeric list> the calibration coefficient array
<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>Related Command</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<i>Parameter</i>	<port> the number of the port = 1
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>



## 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 }
<i>Parameter</i>	<port> the number of the port = 1
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COEF:METH:SOLT1

SENSe&lt;Ch&gt;:CORRection:COEFFicient:METhod:SOLT1 &lt;port&gt;

<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 }
<i>Parameter</i>	<port> the number of the port = 1
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Related Commands</i>	SENS:CORR:COEF:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COEF:SAVE

SENSe&lt;Ch&gt;: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 &lt;Ch&gt;, &lt;Ch&gt;={ [ 1 ]   2   3   4 }</p>
<i>Related Commands</i>	<p>Calibration type selection: SENS:CORR:COEF:METH:SOLT1 SENS:CORR:COEF:METH:OPEN SENS:CORR:COEF:METH:SHOR</p> <p>Calibration coefficient writing: SENS:CORR:COEF</p>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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. Error code: 222.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Calibration Kit &gt; Cal Kit n</b>

## 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 }
<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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Label</b>

## SENS:CORR:COLL:CKIT:RES

SENSe&lt;Ch&gt;: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 }
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Restore</b>

## SENS:CORR:COLL:CKIT:STAN:C0

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;:C0 &lt;numeric&gt;

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;: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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; C0 10–15 F</b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; C1 10–27 F/Hz</b>

## SENS:CORR:COLL:CKIT:STAN:C2

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;:C2 &lt;numeric&gt;

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;: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 } <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 <sup>2</sup> (Farad/Hertz <sup>2</sup> )
<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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; C2 10-36 F/Hz<sup>2</sup></b>



## 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 } <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 <sup>3</sup> (Farad/Hertz <sup>3</sup> )
<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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; C3 10-45 F/Hz<sup>3</sup></b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Offset Delay</b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; L0 10-12 H</b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; L1 10-24 H/Hz</b>

## 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 } <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/Hz2 (Henry/Hertz2)
<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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; L2 10-33 H/Hz2</b>

## 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 } <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 <sup>3</sup> (Henry/Hertz <sup>3</sup> )
<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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; L3 10-42 H/Hz<sup>3</sup></b>

## SENS:CORR:COLL:CKIT:STAN:LAB?

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;: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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Label</b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Offset Loss</b>



## SENS:CORR:COLL:CKIT:STAN:TYPE?

SENSe&lt;Ch&gt;:CORRection:COLLect:CKIT:STAN&lt;Std&gt;: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 } <Std>={ [1]   2   . . . N}, where N – the number of the standards in the calibration kit
<i>Parameter</i>	OPEN : Open SHORT : Short LOAD : Load THRU : Thru
<i>Out of Range</i>	Error occurs. The command is ignored. Error code: 216.
<i>Query Response</i>	{ OPEN   SHOR   LOAD   THRU }
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Standard Type</b>

## 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 } <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>	<b>Calibration &gt; Calibration Kit &gt; Edit Cal Kit &gt; Offset Z0</b>

## 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 }
<i>Equivalent Softkeys</i>	<b>Calibration &gt; { Open   Short   Load } &gt; Cancel</b>

## SENS:CORR:COLL:ECAL:CHECK:EXEC

SENSe&lt;Ch&gt;:CORRection:COLLect:ECAL:CHECK:EXECute

<i>Description</i>	Executing confidence check
<i>Target</i>	AutoCal module (Ch - arbitrary number 1 to 4)
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Autocalibration &gt; Confidence Check</b>

## SENS:CORR:COLL:ECAL:ORI:EXEC

SENSe:CORRection:COLLect:ECAL:ORIEntation:EXECute

<i>Description</i>	Executing AutoCal orientation
<i>Target</i>	AutoCal module
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Autocalibration &gt; Perform Auto-Orientation</b>

## 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 Autocalibration (command/query).
<i>Target</i>	Autocalibration
<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>	<b>Calibration &gt; Autocalibration &gt; Orientation &gt; Auto-Orientation</b>

## SENS:CORR:COLL:ECAL:SOLT1

SENSe&lt;Ch&gt;:CORRection:COLLect:ECAL:SOLT1 &lt;numeric&gt;

<i>Description</i>	Executes 1-port calibration of the specified port of specified channel (Ch) using the Autocalibration module (command only).
<i>Target</i>	Channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Parameter</i>	<numeric> : Port Number = 1
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Autocalibration &gt; Calibrate</b>

## SENS:CORR:COLL:ECAL:SOLT2

SENSe&lt;Ch&gt;:CORRection:COLLect:ECAL:SOLT2 &lt;numeric1&gt;,&lt;numeric2&gt;

<i>Description</i>	Executes the one path 2-port calibration in both directions of the specified ports of selected channel (Ch) using the AutoCal module. (in multiport mode only)  Data(0)            the number of the receiver port;  Data(1)            the number of the source port
<i>Target</i>	Channel <Ch>, <Ch>={ [1] ...   16 } (in multiport mode only).
<i>Parameter</i>	Port number from 1 to 16. The array elements can not contain the same port numbers.
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Autocalibration &gt; Calibrate</b>

## 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 Autocalibration (factory or user characterization). (command/query).
<i>Target</i>	Autocalibration
<i>Parameter</i>	<p>&lt;char&gt; 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>	<b>Calibration &gt; Autocalibration &gt; Characterization</b>

## SENS:CORR:COLL:LOAD

SENSe&lt;Ch&gt;:CORRection:COLLect[:ACQuire]:LOAD &lt;numeric&gt;

<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 } <Ch>={ [1]...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<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>	<b>Calibration &gt; Load</b>

## SENS:CORR:COLL:METH:ERES

```
SENSe<Ch>:CORRection:COLLect:METHod:ERESponse <numeric1>,
<numeric2>
```

<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 (Ch) :CORR:COLL:SAVE command. (in multiport mode only)
<i>Target</i>	Channel <Ch>, <Ch>={ [ 1 ] ...   16 } (in multiport mode only).
<i>Parameter</i>	Port number from 1 to 16. The array elements can not contain the same port numbers.
<i>Out of Range</i>	Error occurs.
<i>Related commands</i>	SENS :CORR :COLL :SAVE
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Calibration Type</b>

## SENS:CORR:COLL:METH:OPEN

SENSe&lt;Ch&gt;:CORRection:COLLect:METhod[:RESPonse]:OPEN &lt;numeric&gt;

<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 } <Ch>={ [1] ...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>



## SENS:CORR:COLL:METH: SHOR

SENSe&lt;Ch&gt;:CORRection:COLLect:MEthod[:RESPonse]:SHORT &lt;numeric&gt;

<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 } <Ch>={ [1] ...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COLL:METH:SOLT1

SENSe&lt;Ch&gt;:CORRection:COLLect:METhod:SOLT1 &lt;port&gt;

<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 }
<i>Parameter</i>	<port> the number of the port = 1
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COLL:METH: THRU

```
SENSe<Ch>:CORRection:COLLect:METhod[:RESPonse]:THRU
<numeric1>, <numeric2>
```

<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 } <Ch>={ [1] ...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<i>Out of Range</i>	Error occurs. The command is ignored.
<i>Related Commands</i>	SENS:CORR:COLL:SAVE
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COLL:METH:TYPE?

SENSe&lt;Ch&gt;: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 }
<i>Query Response</i>	RESPO : Response (Open) RESPS : Response (Short) SOLT1 : Full 1-port calibration NONE : Not defined
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:CORR:COLL:OPEN

SENSe&lt;Ch&gt;:CORRection:COLLect[:ACQuire]:OPEN &lt;numeric&gt;

<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 } <Ch>={ [1]...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<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>	<b>Calibration &gt; Open</b>

## SENS:CORR:COLL:SAVE

SENSe&lt;Ch&gt;:CORRection:COLLect:SAVE

<i>Description</i>	<p>Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.</p> <p>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 &lt;Ch&gt;, &lt;Ch&gt;={ [ 1 ]   2   3   4 }</p>
<i>Related Commands</i>	<p>Calibration type selection: SENS:CORR:COLL:METH:OPEN SENS:CORR:COLL:METH:SHOR SENS:CORR:COLL:METH:SOLT1</p> <p>Calibration standards measurement: SENS:CORR:COLL:LOAD SENS:CORR:COLL:OPEN SENS:CORR:COLL:SHOR</p>
<i>Equivalent Softkeys</i>	<p><b>Calibration &gt; { Open   Short   Load } &gt; Apply</b></p>

## SENS:CORR:COLL:SHOR

```
SENSe<Ch>:CORRection:COLLect[:ACQuire]:SHORT <numeric>
```

<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 } <Ch>={ [1]...   16 } (in multiport mode only).
<i>Parameter</i>	Port number is 1, 1-16 (in multiport mode only).
<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>	<b>Calibration &gt; Short</b>

## SENS:CORR:COLL:THRU

```
SENSe<Ch>:CORRection:COLLect[:ACQuire]:THRU <numeric1>,
<numeric2>
```

<i>Description</i>	Measures the calibration data of the thru standard between the source port and the receiver port. (in multiport mode only)
<i>Target</i>	Channel <Ch>, <Ch>={ [1]... 16} (in multiport mode only).
<i>Range</i>	Port number is 1-16
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	<b>Calibration &gt; Thru</b>

## 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 }
<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>	<b>Calibration &gt; Port Extension &gt;Port Extension</b>



## 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 } <Pt>={ [1] }
<i>Parameter</i>	<frequency> the frequency value from 85E6 to 5.4E9 (R54), from 85E6 to 14E9 (R140), from 1e6 to 6e9 (R60), from 1e6 to 18e9 (R180)
<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>	<b>Calibration &gt; Port Extension &gt; { Freq1   Freq2 }</b>

## 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 }
<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>	<b>Calibration &gt; Port Extension &gt; { Loss1   Loss2 }</b>

## 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 } <Pt>={ [1] }
<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>	<b>Calibration &gt; Port Extension &gt; Loss at DC</b>

## 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 } <Pt>={ [1] }
<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>	<b>Calibration &gt; Port Extension &gt; { Loss1   Loss2 }</b>

## 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 } <Pt>={ [1] }
<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>	<b>Calibration &gt; Port Extension &gt; Port Extension</b>

## 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>	<b>None</b>

## 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 }
<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>	<b>Calibration &gt; Correction</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Cable Loss Correction &gt; Frequency</b>

## 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 }
<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>	<b>Analysis &gt; Time Domain &gt; Cable Loss Correction &gt; Loss</b>



## 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 }
<i>Parameter</i>	<numeric> the cable velocity factor.
<i>Query Response</i>	<numeric>
<i>Preset Value</i>	1.0
<i>Equivalent Softkeys</i>	<b>Analysis &gt; Time Domain &gt; Cable Loss Correction &gt; Velocity Factor</b>

## SENS:CORR:TRAN:TIME:STAT

SENSe&lt;Ch&gt;:CORRection:TRANsform:TIME:STATe &lt;bool&gt;

SENSe&lt;Ch&gt;: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 }
<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>	<b>Analysis &gt; Time Domain &gt; Cable Loss Correction &gt; Cable Loss Correction</b>

## SENS:CORR:TYPE?

SENSe&lt;Ch&gt;:CORRection:TYPE&lt;Tr&gt;?

<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 } <Ch>={ [1]   2   3   4 }
<i>Query Response</i>	{ RESPO   RESPS   SOLT1   NONE }, <srcport>, <rcvport> Where: RESPO : Response (Open) RESPS : Response (Short) SOLT1 : Full 1-port calibration NONE : Not defined  <rcvport>, the number of the receiver port = 1 <srcport>, the number of the source port = 1
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:FREQ:DATA?

SENSe&lt;Ch&gt;: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>&lt;numeric n&gt; the frequency value at the n-th measurement point</p>
<i>Target</i>	<p>Channel &lt;Ch&gt;,          &lt;Ch&gt;={ [1]  2 3 4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric N&gt;</p> <p>The data transfer format depends on the FORM:DATA command setting.</p>
<i>Related Commands</i>	FORM:DATA
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<i>Parameter</i>	<frequency> the stimulus center value from 85E6 to 5.4E9 ( R54), from 85E6 to 14E9 (R140), from 1e6 to 6e9 (R60), from 1e6 to 18e9 (R180)
<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>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<i>Parameter</i>	<frequency> the stimulus span value from 0 to 5.315E9 (R54), from 0 to 13.915E9 (R140), from 0 to 5.999e9 (R60), from 0 to 17.999e9 (R180).
<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>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<i>Parameter</i>	<frequency> the stimulus start value from 85E6 to 5.4E9 (R54), from 85E6 to 14E9 (R140), from 1e6 to 6e9 (R60), from 1e6 to 18e9 (R180).
<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>Equivalent Softkeys</i>	<b>Stimulus &gt; Start Frequency</b>

## 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 }
<i>Parameter</i>	<frequency> the stimulus start value from 85E6 to 5.4E9 (R54), from 85E6 to 14E9 (R140), from 1e6 to 6e9 (R60), from 1e6 to 18e9 (R180).
<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>Equivalent Softkeys</i>	<b>Stimulus &gt; Stop Frequency</b>

## 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 }
<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. Error code: 224.
<i>Query Response</i>	{ INT   EXT }
<i>Preset Value</i>	INT
<i>Equivalent Softkeys</i>	<b>System &gt; Reference Source</b>



## 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>{ &lt;Buf&gt;, &lt;Flag1&gt;, &lt;Flag2&gt;, &lt;Flag3&gt;, &lt;Flag4&gt;, &lt;Flag5&gt;, &lt;N&gt;, &lt;Start 1&gt;, &lt;Stop 1&gt;, &lt;NOP 1&gt; [,&lt;IFBW 1&gt;] [,&lt;Pow 1&gt;] [,&lt;Del 1&gt;] [,&lt;Time 1&gt;], &lt;Start 2&gt;, &lt;Stop 2&gt;, &lt;NOP 2&gt; [,&lt;IFBW 2&gt;] [,&lt;Pow 2&gt;] [,&lt;Del 2&gt;] [,&lt;Time 2&gt;], ... &lt;StartN&gt;, &lt;StopN&gt;, &lt;NOP N&gt; [,&lt;IFBW N&gt;] [,&lt;Pow N&gt;] [,&lt;Del N&gt;] [,&lt;TimeN&gt;] }</pre> <p>&lt;Buf&gt; Always 5,  &lt;Flag1&gt; Stimulus start setting (0 – start/stop, 1 – center/span),  &lt;Flag2&gt; Setting of the &lt;IFBW&gt; field (0 – disabled, 1 – enabled),  &lt;Flag3&gt; Setting of the &lt;Pow&gt; field (0 – disabled, 1 – enabled),  &lt;Flag4&gt; Setting of the &lt;Del&gt; field (0 – disabled, 1 – enabled),  &lt;Flag5&gt; Setting of the &lt;Time&gt; field (0 – disabled, 1 – enabled),  &lt;N&gt; Number of segments,  &lt;Start n&gt; Start value of the n–th segment,  &lt;Stop n&gt; Stop value of the n–th segment,  &lt;NOP n&gt; Number of points of the n–th segment,  &lt;IFBW n&gt; IF bandwidth of the n–th segment (if enabled),  &lt;Pow n&gt; Power of the n–th segment (if enabled),  &lt;Del n&gt; Measurement delay of the n–th segment (if enabled),  &lt;Time n&gt; Reserved for future use (if enabled)</p>
<i>Target</i>	<p>Channel &lt;Ch&gt;,  &lt;Ch&gt;={ [1]   2   3   4 }</p>
<i>Query Response</i>	<p>&lt;numeric 1&gt;, &lt;numeric 2&gt;, ...&lt;numeric 7+M×N&gt;</p> <p>Where,  N – the number of the segments,  M – depends on the values of the flags  M = 3 + &lt;Flag2&gt; + &lt;Flag3&gt; + &lt;Flag4&gt; + &lt;Flag5&gt;</p>
<i>Equivalent Softkeys</i>	<b>None</b>

## SENS:SWE:POIN

SENSe&lt;Ch&gt;:SWEep:POINTs &lt;numeric&gt;

SENSe&lt;Ch&gt;: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 }
<i>Parameter</i>	<numeric> the number of measurement points from 2 to 100001.
<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>	<b>Stimulus &gt; Points</b>

## 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 }
<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>	<b>None</b>

## 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 }
<i>Parameter</i>	<char> specifies the sweep type: LINear : Linear frequency sweep LOGarithmic : Logarithmic frequency sweep SEGMENT : Segment frequency sweep
<i>Out of Range</i>	Error occurs. The command is ignored. Error code: 206.
<i>Query Response</i>	{ LIN   LOG   SEGM }
<i>Preset Value</i>	LIN
<i>Equivalent Softkeys</i>	<b>Stimulus &gt; Sweep Type</b>

## 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 4
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>None</b>

## SERV:CHAN:TRAC:ACT?

SERVice:CHANnel&lt;Ch&gt;: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 }
<i>Query Response</i>	<numeric> from 1 to 4
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## SOUR:POW

```
SOURce<Ch>:POWER[:LEVel][:IMMediate][:AMPLitude]
{MIN|MAX|<power>}
```

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

<i>Description</i>	Sets or reads out the power level for the frequency sweep (command/query). (only for R60, R180)
<i>Target</i>	Channel <Ch>, <Ch>={ [1]  2 3 4 5 6 7 8 9 }
<i>Parameter</i>	<power> level power
<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>	<b>Stimulus &gt; Power &gt; Output Power</b>



## SOUR:POW:STAT

SOURce<Ch>:POWer[:LEVel]:STATe <char>

<i>Description</i>	Sets or reads out the power level for the frequency sweep. (only for R54/R140).
<i>Target</i>	Channel <Ch>, <Ch>={ [1]   2   3   4 }
<i>Parameter</i>	<char> specifies the power level for the frequency sweep:  HIGH : high output power;  LOW : low output power:  ROFF : RF Out is Off.
<i>Out of Range</i>	Error occurs. The command is ignored..
<i>Query Response</i>	{ HIGH   LOW   ROFF }
<i>Preset Value</i>	"HIGH"
<i>Equivalent Softkeys</i>	<b>Stimulus &gt; Power</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## STAT:QUES:COND?

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

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

## STAT:QUES:ENAB

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

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

<i>Description</i>	Sets or reads out the value of the <u>Questionable Status Enable Register</u> (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>	<b>None</b>

## 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 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>None</b>



## 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 }
<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>	<b>None</b>

## 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 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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 }
<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>	<b>None</b>

## 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 }
<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>	<b>None</b>



## 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 }
<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>	<b>None</b>

## 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 }
<i>Query Response</i>	<numeric>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>None</b>

## 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>	<b>System &gt; System Correction</b>

## 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>	<b>None</b>

## 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>&lt;numeric&gt;, &lt;string&gt; Where:</p> <p>&lt;numeric&gt;        error code</p> <p>&lt;string&gt;         error message</p> <p>If there is no error in the queue, "0, No error" is read out.</p>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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</p> <p>SYST:RWL</p>
<i>Equivalent Softkeys</i>	<b>None</b>

## 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>	<b>System &gt; Preset &gt; OK</b>

## 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>	<b>None</b>



## 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>	<b>None</b>

## 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>	<b>None</b>

## SYST:TEMP:SENS?

SYSTem:TEMPerature:SENSor? &lt;numeric&gt;

<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>	<b>None</b>

## 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>	<numeric 1> hours from 0 to 23; <numeric 2> minutes from 0 to 59; <numeric 3> seconds from 0 to 59.
<i>Query Response</i>	<numeric 1>, <numeric 2>, <numeric 3>
<i>Equivalent Softkeys</i>	<b>None</b>

## TRIG

```
TRIGger[:SEquence][:IMMEDIATE]
```

<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the External). If the trigger source is set to External, an error occurs (error code 221) 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 Hold), an error occurs (error code 211) and the command is ignored.</p> <p>The command is completed before the end of the sweep. (no query)</p>
<i>Related Commands</i>	TRIG:SOUR INIT:CONT INIT:IMM
<i>Equivalent Softkeys</i>	<b>None</b>

## TRIG:EXT:DEL

```
TRIGger[:SEQuence]:EXTernal:DELay <time>
```

```
TRIGger[:SEQuence]:EXTernal:DELay?
```

<i>Description</i>	Sets or reads out the response delay with respect to the external trigger signal. (command/query)
<i>Parameter</i>	<time> the delay value from 0 to 100 sec.
<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>Related Commands</i>	TRIG:SOUR EXT
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Input &gt; Delay</b>

## TRIG:EXT:POS

```
TRIGger[:SEQuence]:EXTernal:POSition <char>
```

```
TRIGger[:SEQuence]:EXTernal:POSition?
```

<i>Description</i>	<p>Selects the position of the external trigger. The Analyzer waits for external trigger:</p> <ul style="list-style-type: none"> <li>• Before sampling, when the frequency of the stimulus port have been set.</li> <li>• Before the frequency setup and subsequent measurement. The frequency change of the stimulus port begins when the external trigger arrives.</li> </ul> <p>Depending on the command TRIG:SEQ:POIN the external trigger wait occurs before each point or before the first point of the full sweep cycle. (command/query)</p>
<i>Parameter</i>	<p>&lt;char&gt; Choose from:</p> <p>BSAM : Before sampling</p> <p>BSET" : Before frequency setup</p>
<i>Query Response</i>	{ BSAM   BSET }
<i>Preset value</i>	BSAM
<i>Related Commands</i>	TRIG:SOUR
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Input &gt; Position &gt; Before Sampling   Before Setup</b>

## TRIG:EXT:SLOP

```
TRIGger[:SEquence]:EXternal:SLOPe <char>
```

```
TRIGger[:SEquence]:EXternal:SLOPe?
```

<i>Description</i>	Sets or reads out the polarity of the external trigger. (command/query)
<i>Parameter</i>	<char> Choose from:  POSitive : Positive edge  NEGative : Negative edge
<i>Query Response</i>	{ POS   NEG }
<i>Preset value</i>	NEG
<i>Related Commands</i>	TRIG:SOUR
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Input&gt; Polarity &gt; Negative Edge   Positive Edge</b>

## TRIG:OUTP:FUNC

```
TRIGger:OUTPut:FUNction <char>
```

```
TRIGger:OUTPut:FUNction?
```

<i>Description</i>	Selects the trigger output function. The trigger output outputs various waveforms depending on the setting of the Output Trigger Function (see the operating manual). (command/query)
<i>Parameter</i>	<p>&lt;char&gt; Choose from:</p> <p>BSET : Before frequency setup pulse</p> <p>BSAM : Before sampling pulse</p> <p>ASAM : After sampling pulse</p> <p>RTRG : Ready for trigger signal</p> <p>ESWP : End of sweep pulse</p> <p>MEAS : Measurement sweep signal</p>
<i>Query Response</i>	{ BSET   BSAM   ASAM   RTGR   ESWP   MEAS }
<i>Preset value</i>	RTRG
<i>Related Commands</i>	TRIG:OUTP:STAT
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Output &gt; Position &gt; Before Setup   Before Sampling   After Sampling   Ready for Trig   Sweep End   Measurement</b>

## TRIG:OUTP:POL

```
TRIGger:OUTPut:POLarity <char>
```

```
TRIGger:OUTPut:POLarity?
```

<i>Description</i>	Sets or reads out the polarity of the trigger output. (command/query)
<i>Parameter</i>	<char> Choose from:  POSitive : Positive edge  NEGative : Positive edge
<i>Query Response</i>	{ POS   NEG }
<i>Preset value</i>	NEG
<i>Related Commands</i>	TRIG:OUTP:FUNC
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Output &gt; Polarity &gt; Negative Edge   Positive Edge</b>



## TRIG:OUTP:STAT

```
TRIGger:OUTPut:STATe {OFF|ON|0|1}
```

```
TRIGger:OUTPut:STATe?
```

<i>Description</i>	Turns ON/OFF the trigger output. (command/query)
<i>Parameter</i>	{ON 1}          ON  {OFF 0}        OFF
<i>Query Response</i>	{0 1}
<i>Preset value</i>	0
<i>Related Commands</i>	TRIG:OUTP:FUNC
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Output &gt; Enable Out&gt; ON   OFF</b>

## TRIG:POIN

```
TRIGger[:SEquence]:POINT {OFF|ON|0|1}
```

```
TRIGger[:SEquence]:POINT?
```

<i>Description</i>	Turns ON/OFF the point trigger feature. When the point trigger is turned ON, the external trigger response is the single point. When the point trigger feature is turned OFF, the external trigger response is the entire sweep. (command/query)
<i>Parameter</i>	{ON 1}          ON  {OFF 0}        OFF
<i>Query Response</i>	{0 1}
<i>Preset value</i>	0
<i>Related Commands</i>	TRIG:SOUR
<i>Equivalent Softkeys</i>	<b>Trigger &gt; Trigger Input &gt; Event &gt; On Sweep   On Point</b>

## TRIG:SING

TRIGger[:SEQuence]:SINGle

<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the External). If the trigger source is set to External, an error occurs (error code 221) 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 Hold), an error occurs (error code 211) 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>	<b>None</b>

## TRIG:SOUR

```
TRIGger[:SEQuence]:SOURce <char>
```

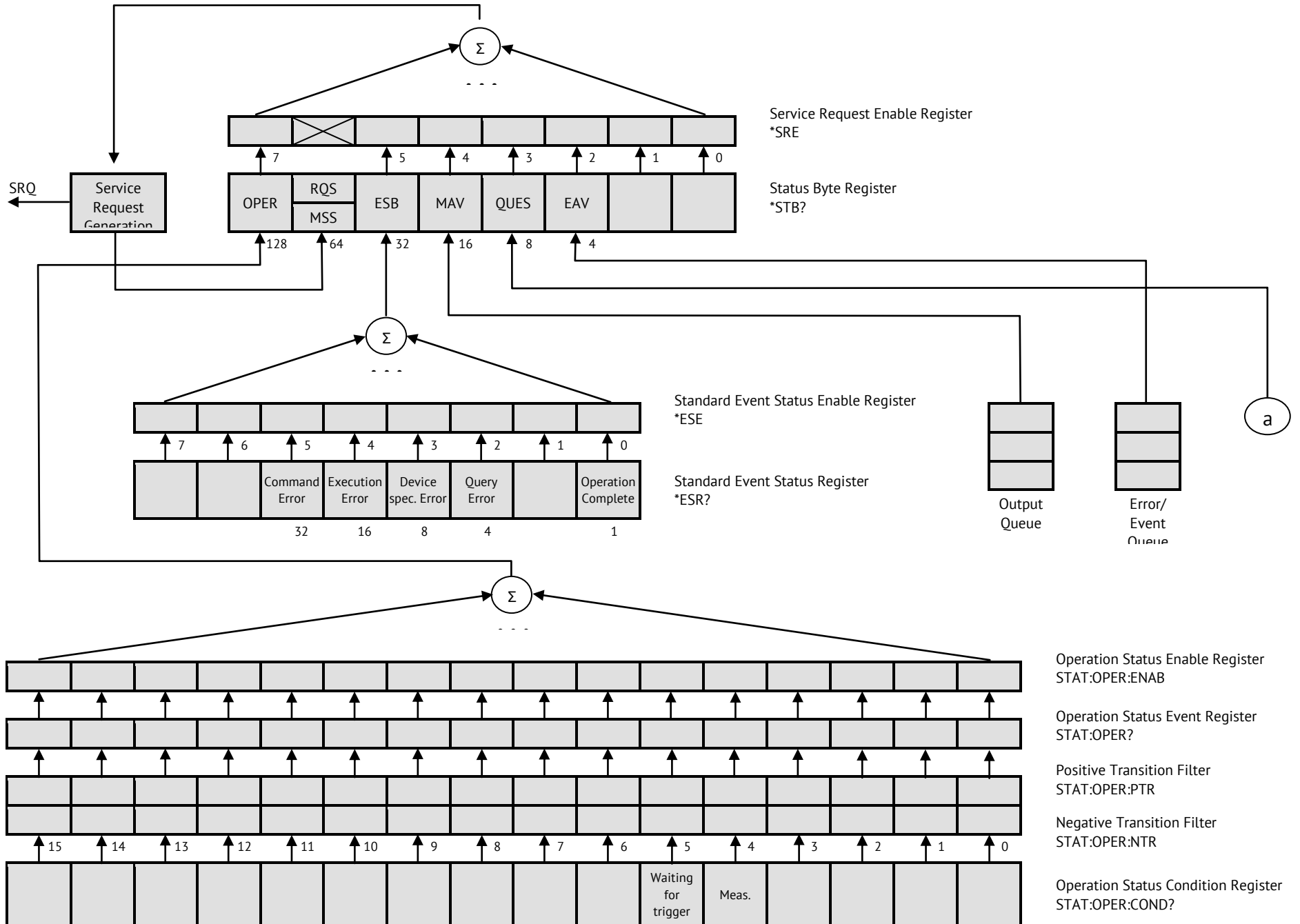
```
TRIGger[:SEQuence]:SOURce?
```

<i>Description</i>	Sets or reads out the sweep trigger source (command/query).
<i>Parameter</i>	<p>&lt;char&gt; choose from:</p> <p>INTernal : Internal</p> <p>EXTernal : External</p> <p>BUS : Bus</p>
<i>Out of Range</i>	Error occurs. The command is ignored. Error code: 205
<i>Query Response</i>	{ INT   EXT   MAN   BUS }
<i>Preset Value</i>	INT
<i>Related Commands</i>	TRIG TRIG:SING *TRG
<i>Equivalent Softkeys</i>	<b>Trigger&gt; Trigger Source &gt; { Internal   External   Bus }</b>

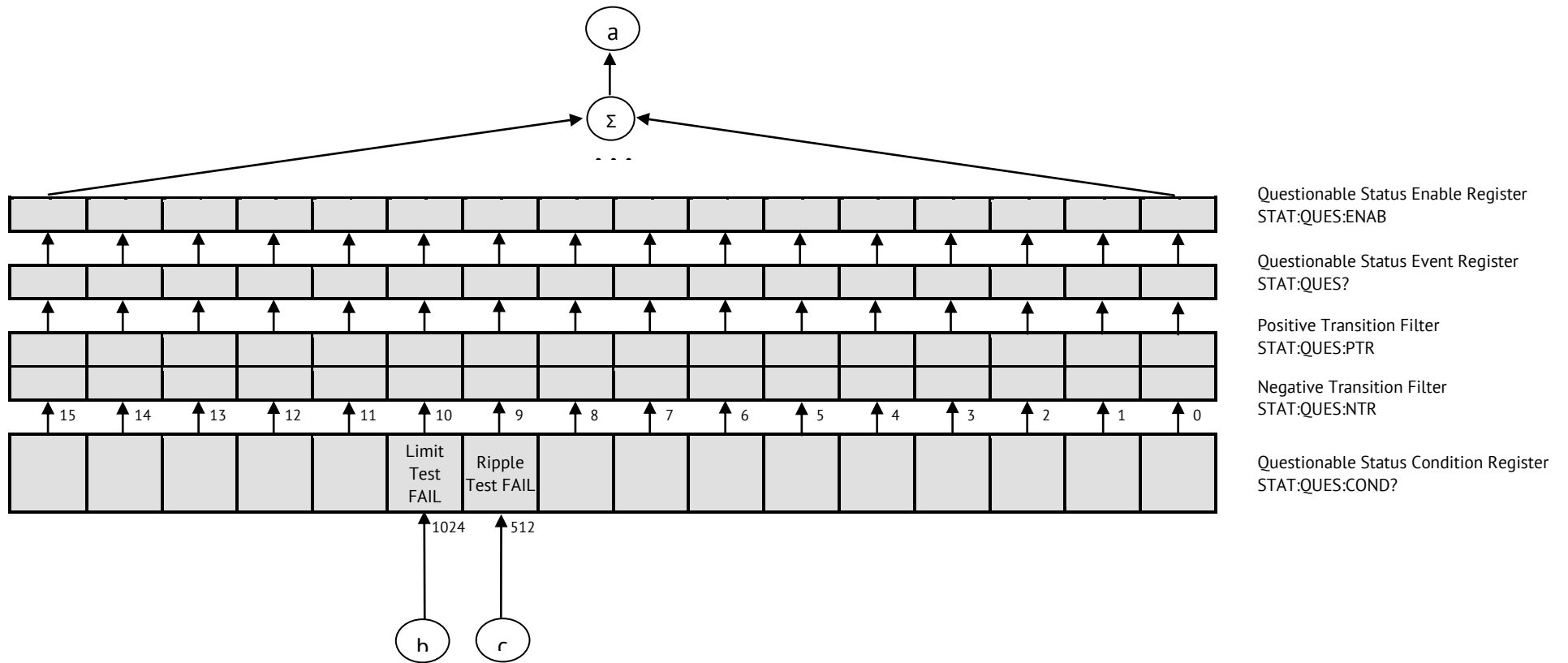
## TRIG:WAIT

TRIGger[:SEquence]:WAIT &lt;char&gt;

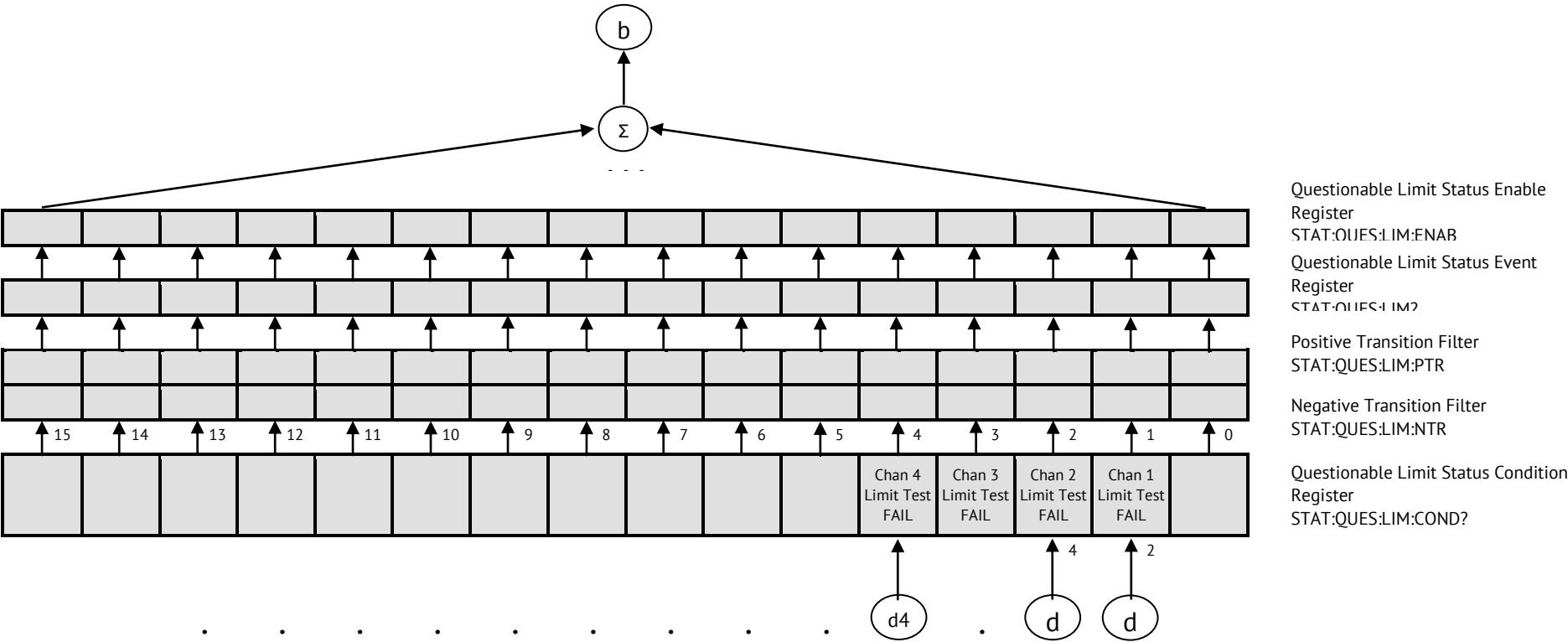
<i>Description</i>	<p>Delays the execution by the analyzer of the next command till the specified state of the analyzer has been reached (see options below).</p> <p>A query that follows the TRIG:WAIT command blocks the execution of the user program till the specified state of the analyzer has been reached.</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>&lt;char&gt; 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><b>None</b></p>



Appendix 1. IEE488.2 Status Reporting System

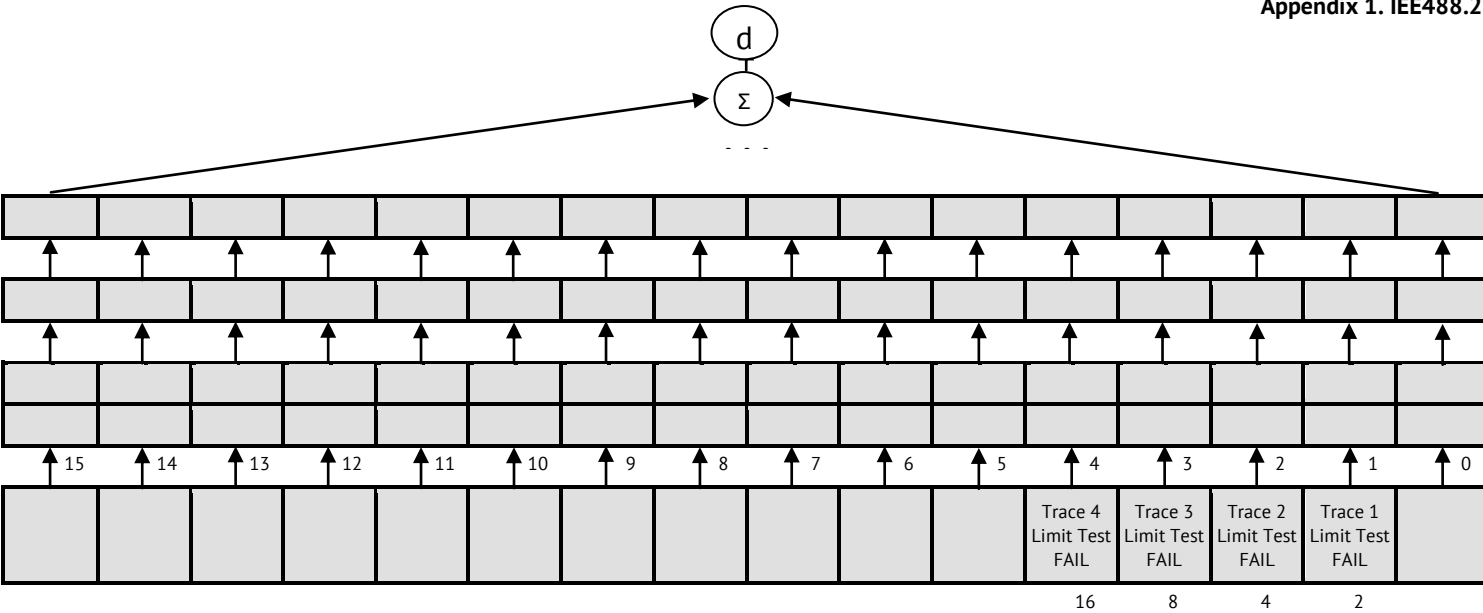


Appendix 1. IEE488.2 Status Reporting System





Appendix 1. IEE488.2 Status Reporting System



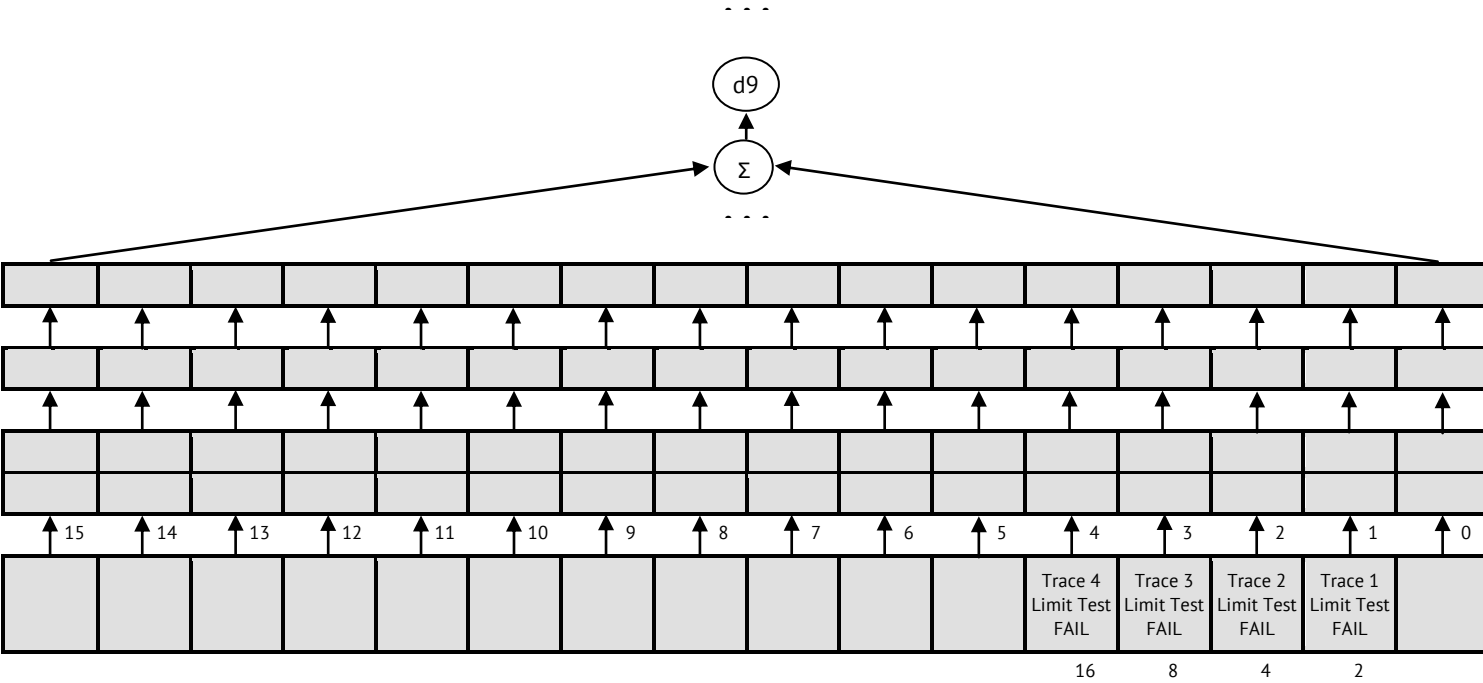
Questionable Limit Channel 1 Status Enable Register  
STAT:QUES:LIM:CHAN1:ENAB

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

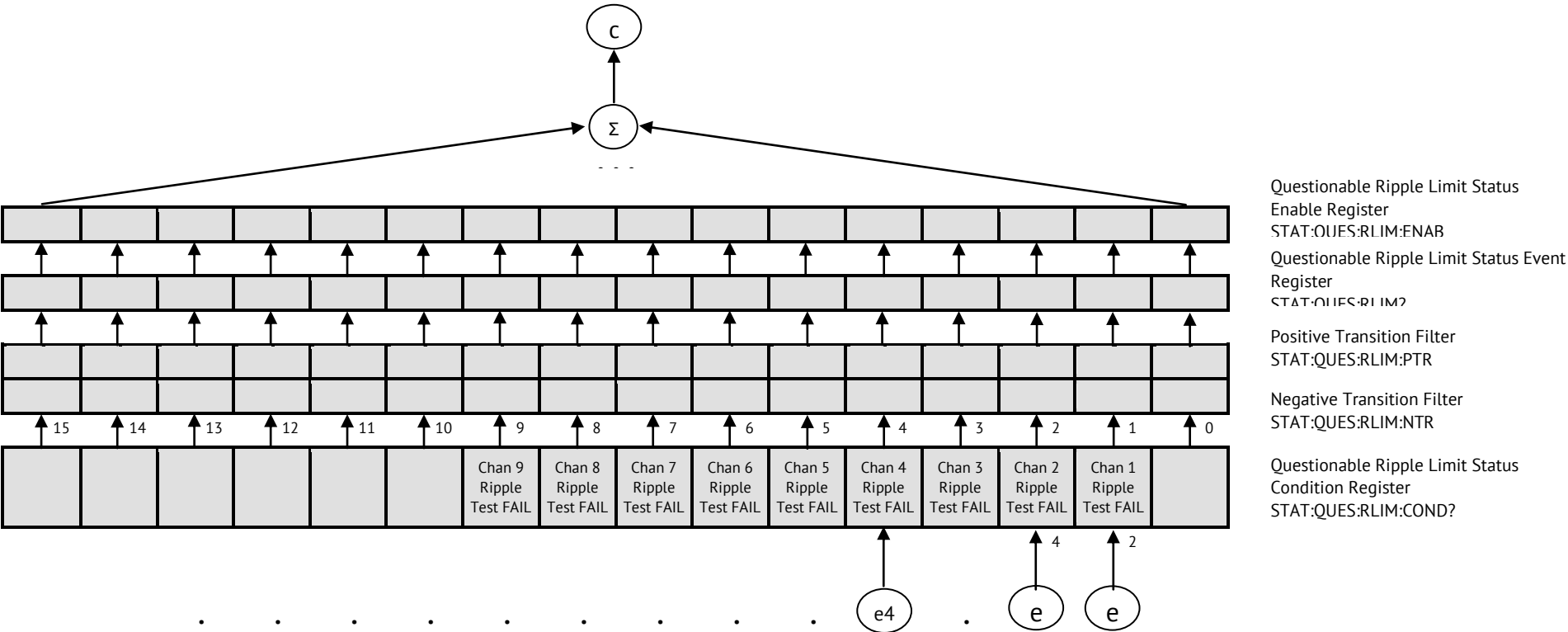
Questionable Limit Channel 9 Status Event Register  
STAT:QUES:CHAN9:IM?

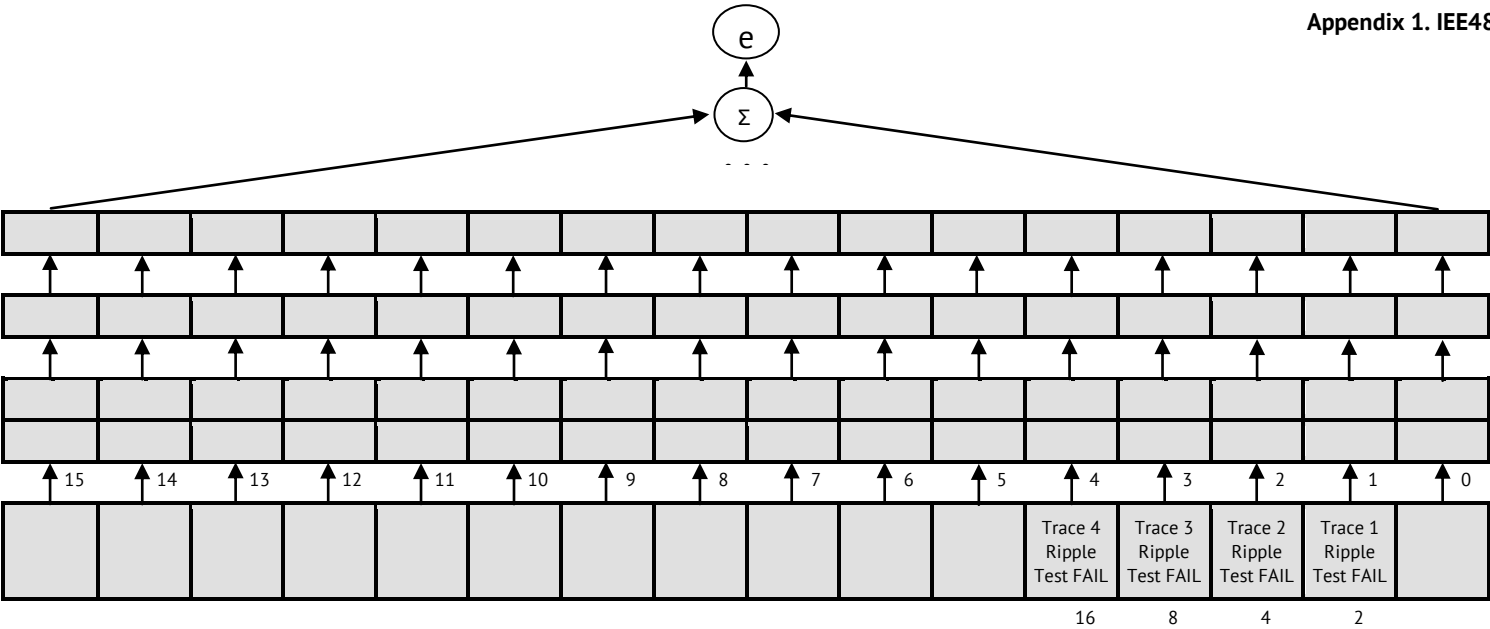
Positive Transition Filter  
STAT:QUES:LIM:CHAN9:PTR

Negative Transition Filter  
STAT:QUES:LIM:CHAN9:NTR

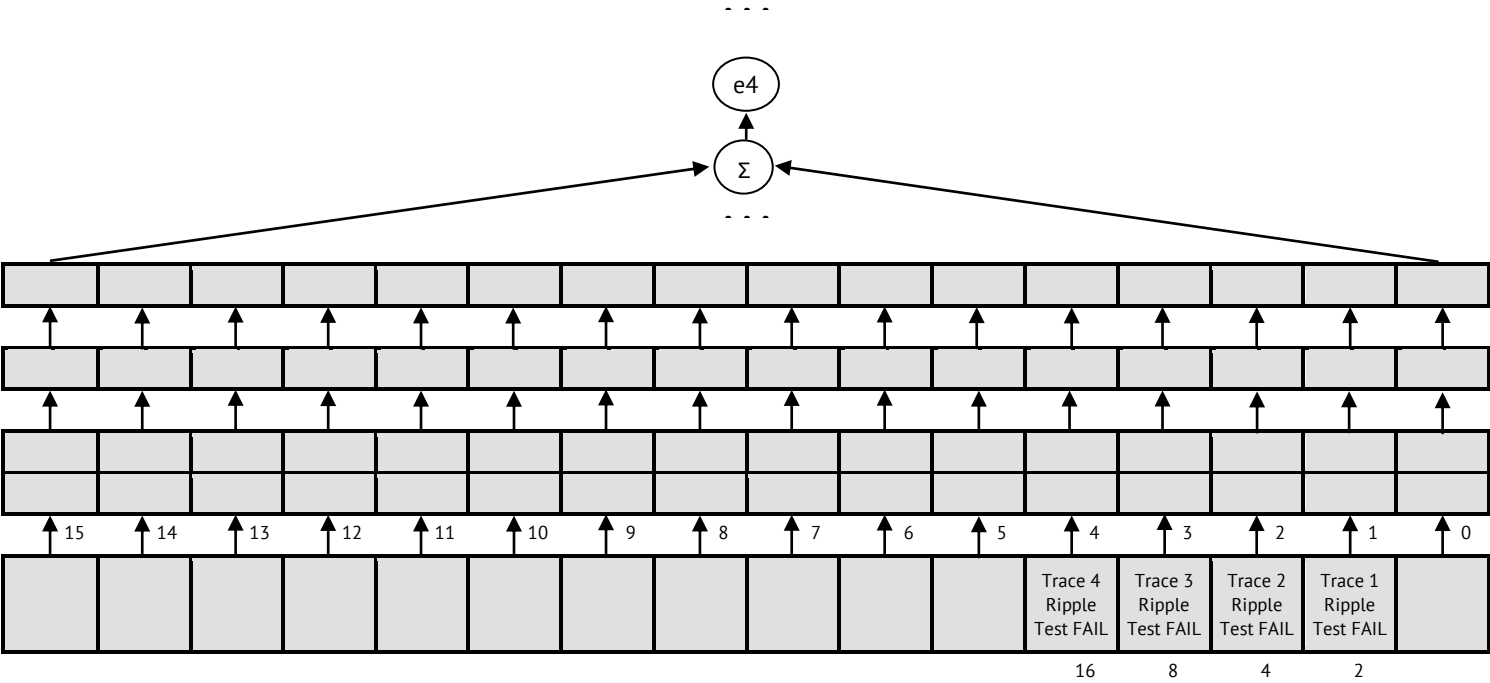
Questionable Limit Channel 9 Status Condition Register  
STAT:QUES:LIM:CHAN9:COND?

Appendix 1. IEE488.2 Status Reporting System





Questionable Ripple Limit Channel 1  
 Status Enable Register  
 STAT:QUES:RLIM:CHAN1:ENAB  
 Questionable Ripple Limit Channel 1  
 Status Event Register  
 STAT:QUES:CHAN1:RIM?  
 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:RIM?  
 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;                // Error checking
    ViSession defaultRM, instr;     // Communication channels
    ViUInt32 retCount;             // Return count from string I/O
    ViByte buffer[255];           // Buffer for string I/O
    int temp;
    int NOP = 21;                  // Number of measurement points
    const int maxCnt = 100;        // Maximum reading count
    double Data[maxCnt*2];        // Measurement data array
    double Freq[maxCnt];          // Frequency array

    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");
    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:FORM SWR\n");
viPrintf(instr, "SENS:BAND 1000\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.

