



PicoScope 9000 Series PC Sampling Oscilloscopes

Programmer's Guide



Contents

1 PicoScope 9000 API Reference	1
1 PicoScope9000 COM Server	1
2 ExecCommand Method	1
3 COMRC Object	1
2 Commands Syntax	2
1 Command and Query Structure	2
1 Overview	2
2 Messages	2
3 Commands	3
4 Queries	3
5 Headers	3
2 Command Entry	4
1 Rules	4
2 Concatenating	4
3 Command Classification	6
1 Execution-type commands	6
2 On/off-type commands	6
3 On/off-group-type commands	6
4 Selector-type commands	8
5 Integer-type commands	8
6 Float-type commands	8
7 Data-type commands	9
4 Full list of commands	10
1 Header commands	10
2 GUI commands	10
3 System commands	10
4 Channels commands	11
5 Timebase commands	13
6 Trigger commands	14
7 Acquisition commands	16
8 Display commands	18
9 Save/Recall commands	20
1 Work with Memo Zones (M1, M2, M3, M4)	20
2 Work with Disk	21
3 Work with Setups	22
10 Markers commands	23
11 Measure commands	24
1 Measurements of Time Domain Signals	24
2 Measurements of Spectrum Signals	29
3 Getting Measurement Results	31
12 Limit Tests commands	31
13 Mathematics commands	37

14 FFT commands	39
15 Histogram commands	40
1 Setting Histogram Parameters	40
2 Getting Results of Histogram	44
16 Mask Test commands	45
1 Common Mask Commands	45
2 Standard Mask Commands	47
3 Getting Mask Results	48
17 Eye Diagram commands	49
1 Setting Eye Parameters	49
2 Getting Eye Measurement Results	52
18 Utilities commands	53
19 Waveforms commands	55
20 System commands	56
5 Programming Examples	59
1 Delphi	59
2 LabVIEW	59
3 Visual Basic .NET	60
Index.....	61

1 PicoScope 9000 API Reference

PicoScope 9000 provides an API for any third-party application or library to control the oscilloscope and collect signals. The API is *COM-based* and is provided by the PicoScope 9000 GUI application.

ps9000pg.en-4

1.1 PicoScope9000 COM Server

The COM server implementing the API is called *PicoScope9000* and is implemented by the PicoScope 9000 GUI application (*PicoScope9000.exe*). It is registered in the system during the setup process, and can be explicitly unregistered and registered again by executing *PicoScope9000.exe* with the */UnregServer* or */RegServer* switches.

1.2 ExecCommand Method

The COMRC object contains only one method: *ExecCommand*. The method has one argument—a text string with a command or query. The method returns:

- *NULL* (*Nothing* in Visual Basic) if a command without query has been successfully executed
- The text string "ERROR" if the command was invalid
- Another text string with query results if the command was a query or a command with query

The syntax of the commands and query, as well as the full list of commands, is described in the following pages.

1.3 COMRC Object

To implement the API the server exposes only one object, which is called *COMRC*. The object supports automation, so it can be used by high-level languages like JavaScript (HTML pages) or VBA (Microsoft Word). However, low-level languages like C are also supported. The string defining the system-wide name of the object and used for object creation is "*PicoScope9000.COMRC*".

2 Commands Syntax

2.1 Command and Query Structure

2.1.1 Overview

PicoScope 9000 commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and information about its status.

Most commands have both a set form and a query form. The query form of the command differs from the set form by a question mark at the end. For example, the set command:

```
ACquire:Ch1:MODE
```

has a query form:

```
ACquire:Ch1:MODE?.
```

Not all commands have both a set and a query form. Some commands have set only and some have query only.

2.1.2 Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, as defined in the following table.

Command message elements	
Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required.
<Mnemonic>	This is a header of the sub-function. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates item from each other.
<Argument>	This is a quantity, quality, restriction or limit associated with the header. Some commands have no arguments while others have multiple arguments. A space separates arguments from the header. A comma separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and its argument. Optionally, a white space may consist of multiple white space characters.

2.1.3 Commands

Commands cause the instrument to perform a specific function or change one of its settings. Commands have the structure:

```
[ : ]<Header> [ <Space><Argument> [ <Comma><Argument> ] . . . ]
```

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

2.1.4 Queries

Queries cause the instrument to return information about its status or settings. Queries have the structure:

- [:]<Header>?
- [:]<Header>? [<Space><Argument> [<Comma><Argument>] . . .]

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level. For example,

```
HISTogram:STATistics:STDdev?
```

returns the standard deviation of the histogram, while

```
HISTogram:STATistics?
```

returns all the histogram statistics, and

```
HISTogram?
```

returns all the histogram parameters.

2.1.5 Headers

You can control whether the instrument returns headers as part of the query response. Use the HEADer command to control this feature. If header is on, the query response returns command headers and formats itself as a valid set command. When the header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

Comparison of Header Off and Header On Responses		
Query	Header Off	Header On
Ch1:Scale?	200 mV/div	CH1:SCALE 200 mV/div
Acq:Ch1:RecLen?	512	ACQ:CH1:RECLen 512

2.2 Command Entry

2.2.1 Rules

The following rules apply when entering commands:

- You can follow a mnemonic by any letters for more easy understanding of the program's text. For example, the commands:

```
Ch1:ATTEN:DIMENS Volt
```

```
Ch1:ATTENuator:DIMENSION Volt
```

```
Ch1:ATTENblabla:DIMENSblabla Volt
```

are equivalent. However, arguments must be written without any following letter.

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 to 09, and 0B to 20 hexadecimal (0 to 9, and 11 to 32 decimal).
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

2.2.2 Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated commands in the order received. When concatenating commands and queries, you must follow these rules:

- Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands

```
TRIGger:MODE FREE
ACQuire:NUMAVg 10
```

can be concatenated into the following single command:

```
TRIGger:MODE FREE;:ACQuire:NUMAVg 10
```

- If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands:

```
ACQuire:Ch1:MODE ENVMINMAX
ACQuire:Ch1:NAVG 10
```

into a single command:

```
ACQuire:Ch1:MODE ENVMINMAX; NAVG 10
```

The longer version works equally well:

```
ACQuire:CH1:MODE ENVMINMAX;:ACQuire:NAVG 10
```


- Set commands and queries may be concatenated in the same message. For example:

```
ACQuire:CH1:MODE AVGSTAB;NAVG?
```

is a valid message that sets the acquisition mode to Stable Averaging. The message then queries the number of acquisitions for averaging. Concatenated commands and queries are executed in the order received.

- Here are some invalid concatenations:

```
DISPlay:STYLE DOTS;ACQuire:NAVG 10  
(no colon before ACQuire)
```

```
DISPlay:STYLE DOTS;:FORMAT YT  
(extra colon before FORMAT; use DISPlay:STYLE DOTS;FORMAT YT instead)
```

- ```
Acq:Ch1:Mode Sample;Ch1:RecLen 1024
```

  
(levels of the mnemonics are different; either remove the second use of Ch1: or place :Acq: in front of Ch1:

## 3 Command Classification

Most commands can be related to one of a few types. For example, the execution-type commands tell the instrument to perform a specific action; the selector-type commands modify a specific instrument setting to the one of few fixed values, and so on. All commands of a given type have similar behavior.

### 3.1 Execution-type commands

The execution-type commands tell the instrument to perform a specific action. For example:

```
*Run
*ClrDispl
```

There are no arguments for these commands.

All execution-type commands have a 'set' form only, and not a 'query' form.

### 3.2 On/off-type commands

The on/off-type commands tell the instrument turn on or turn off a specific function. For example:

```
Header Off
Ch1:Display 0
```

There are four fixed arguments possible in these commands: `On`, `Off`, `0`, `1`. Arguments `On` and `1` are equivalent and turn on the corresponding function. Arguments `Off` and `0` are also equivalent and turn off the specific function.

All on/off-type commands have a query form. The queries return one of two fixed values: `ON` or `OFF`. It is also possible to use the query form with an argument. For example:

```
Ch1:Display? 0
```

This command turns off the graphic of Channel 1 and returns `OFF`.

### 3.3 On/off-group-type commands

Some functions of the instrument have items that may be set independently on or off. It is also possible for the items to be either all on or all off. An example of this type of command is:

```
Meas:Ch1:XParam
```

This command has a set of parameters for automatic X-axis measurements for Ch1. It is possible to select up to 10 parameters from a list of 18: `Period`, `Freq`, `PosWidth`, `NegWidth`, `Rise`, `Fall`, `PosDuty`, `NegDuty`, `PosCross`, `NegCross`, `BurstWidth`, `Cycles`, `TimeOfMax`, `TimeOfMin`, `PosJitterPp`, `PosJitterRMS`, `NegJitterPp`, `NegJitterRMS`.

There are between 2 and 64 custom items in the on/off-group-type commands. The full set of items is specified for each commands in the [list of commands](#)<sup>[10]</sup>.

The on/off-group-type commands can be used in several modes. Every such command can be used in every mode.

### Single-item mode

Single-item mode is used to control one command's item without changing the other items. In this case the item's mnemonic is added to the end of command after a colon (:) character. This must be followed by a space character and then one of the following arguments: On, Off, 0, 1. For example, the next command turns on a frequency measurement for Channel 1:

```
Meas:Ch1:XParam:Freq 1
```

Single-item mode has a query form similar to the On-Off commands. So, the query:

```
Meas:Ch1:XParam:Period 1
```

or

```
Meas:Ch1:XParam:Freq?
```

returns either ON or OFF.

### Group-on mode

Multi-item mode is used to simultaneously turn on a custom group of items. In this case the `:Include` mnemonic is added to the end of the command. This is then followed by a space and a few items separated by commas (,). For example, the next command turns on rise time and fall time measurements for Channel 1:

```
Meas:Ch1:XParam:Include Rise,Fall
```

### Group-off mode

Multi-item-off mode is used to simultaneously turn off a custom group of items. In this case the `:Exclude` mnemonic is added to the end of the command. This is then followed by a space and a few items separated by commas. For example, the next command turns off frequency and period measurements for Channel 1:

```
Meas:Ch1:XParam:Exclude Freq,Period
```

### All-off mode

All-off mode is used for simultaneously turning off all items. In this case the `:ClearAll` mnemonic is added to the end of the command. For example, the next command turns off all measurements for Channel 1:

```
Meas:Ch1:XParam:ClearAll
```

Group-on, Group-off and All-off modes do not have a query form.

### Group-query mode

Group-query mode is used find out which items are currently turned on. This mode has only the query form. For example:

```
Meas:Ch1:XParam?
```

The answer may be `ClearAll` – if all items are turned off; or one or more items separated by comma (,). For example, the answer `Freq,Period` means two turned on items.

### 3.4 Selector-type commands

The selector-type commands modify a specific instrument setting to one of a few fixed values. For example:

```
Trig:Source
```

has these possible arguments:

```
Direct, ExtHF, IntClock;
```

and

```
Trig:Mode
```

has these possible arguments:

```
Free, Trig
```

Between 2 and 32 custom arguments are available for these commands. The full set of arguments is specified for each command in the [list of commands](#)<sup>[10]</sup>.

The selector-type commands have a query form. It is possible to use the query form with an argument; for example,

```
Trig:Source? Direct
```

This command sets the Direct input as the trigger source and returns `DIRECT`.

### 3.5 Integer-type commands

The integer-type commands modify specific integer-value functions. For example, the command:

```
Acq:Ch1:RecLen 1024
```

sets the length of Channel 1 signals to 1024 points. The valid range and increment of each value is different and is described in the [list of commands](#)<sup>[10]</sup>.

The integer-type commands have a query form. It is possible to use the query form with an argument; for example:

```
Acq:Ch1:RecLen? 24
```

returns 32, since 32 is the minimum valid length of a signal.

### 3.6 Float-type commands

The float-type commands modify specific real-value functions. For example, the command:

```
Ch1: Scale 0.1
```

sets the Y-scale for Channel 1 to 100 mV/div. The valid range and increment of each value is different and is described in the [list of commands](#)<sup>[10]</sup>.

Float-type commands have a query form. It is also possible to use the query form with an argument; for example:

```
Ch1:Scale? 0.1
```

returns 100 mV/div, when V/div is dimension of the scale, and the prefix m is milli.

The commands:

```
TB:ScaleA? 0.0000001
```

```
TB:ScaleA? 100e-9
```

```
TB:ScaleA? 0.1u
```

```
TB:ScaleA? 100p
```

are equivalent and set the Scale A of the timebase to the value 100 ns/div. All of these commands return 100 ns/div.

### 3.7 Data-type commands

The data-type commands are used to send some data to the instrument or to receive some data from the instrument, such as an acquired signal's array of points, the result of a measurement, and so on.

Some data-type commands have a query form only, while others have both a command and a query form. The structure of the data is different for each command and is specified in the [list of commands](#)<sup>[10]</sup>.

## 4 Full list of commands

All of the PicoScope 9000 API commands are listed below, organised by type.

### 4.1 Header commands

Header: `Header`

Type: `On/Off`

Action: `Enable/disable headers as part of the query response.`

### 4.2 GUI commands

Header: `Gui`

Type: `Selector type command`

Arguments: `RemoteLocal, RemoteOnly, Invisible`

Action: `Set the behavior of the GUI when it controls by COM-object`

Version: `This command can be used with PicoScope SW v.2.3.2 or later.`

### 4.3 System commands

#### ***Clear Display***

Header: `*ClrDispl`

Type: `Execution`

Action: `Clear Display immediately.`

#### ***Start Cycle Acquisition***

Header: `*Run`

Type: `Execution`

Action: `Run the instrument`

#### ***Start Single Acquisition / Stop Acquisition***

Header: `*StopSingle`

Type: `Selector`

Arguments: `Stop, Single`

Action: `Single - Start a single acquisition`  
`Stop - Immediately stop the acquisition`

Response: `Stop - the instrument is stopped`  
`Single - the instrument is in the acquisition state`

**Start Autoscaling**

Header: \*Autoscale  
Type: Selector  
Arguments: Auto, SingleVal, NRZ, RZ  
Action: set the type of signal and start autoscaling of the instrument  
Response: selected type of signals.

**Recall Default Setup**

Header: \*DefSetup  
Type: Execution  
Action: Restore the instrument to its Default Setup

**4.4 Channels commands****Display a Channel**

Header: Ch1:Display; Ch2:Display  
Type: On/Off  
Action: turn on or turn off the display of corresponding channel's signal

**Acquire a Channel**

Header: Ch1:Acquire; Ch2:Acquire  
Type: On/Off  
Action: turn on or off the acquisition of the channel's signal when its display is turned off

**Scale a Channel**

Header: Ch1:Scale; Ch2:Scale  
Type: Float  
Argument: 0.002 to 0.5, or other when attenuator is used  
Action: set the specified display scale in V/div

**Offset a Channel**

Header: Ch1:Offset; Ch2:Offset  
Type: Float  
Argument: -1 to +1, or other when attenuator is used  
Action: set the specified compensation voltage of the channel in V

**Bandwidth of Channel**

Header: Ch1:Band; Ch2:Band  
Type: Selector  
Arguments: Full, Narrow  
Action: set the bandwidth of the channel

**Attenuator linear/log**

Header: Ch1:Atten:Unit; Ch2:Atten:Unit  
Type: Selector  
Arguments: Off, Ratio, DB  
Action: set presence and scale of attenuator or converter used with the channel

**Attenuator ratio**

Header: Ch1:Atten:Ratio; Ch2:Atten:Ratio  
Type: Float  
Argument: 0.0001 to 1000000  
Action: set the attenuation ratio. This setting is active only when attenuator unit is 'ratio'

**Attenuator dB**

Header: Ch1:Atten:DB; Ch2:Atten:DB  
Type: Float  
Argument: -80 to +120  
Action: set the attenuation in dB. This setting is active only when attenuator units is decibels.

**Attenuator unit**

Header: Ch1:Atten:Dimens; Ch2:Atten:Dimens  
Type: Selector  
Arguments: Volt, Watt, Ampere, Unknown  
Action: set the units of the converter used with the channel



## 4.5 Timebase commands

### ***Timebase Units***

Header: TB:Units

Type: Selector

Arguments: Time, Bit

Action: set units of timebase to s/div or bit/div

### ***Timebase mode***

Header: TB:Mode

Type: Selector

Arguments: A, AB, B

Action: set main, intensified, or delayed timebase

### ***Main timebase scale, sec/div***

Header: TB:ScaleA

Type: Float

Argument: 10e-12 to 50e-3

Action: set scale of the main timebase when time units are used

### ***Delayed timebase scale, sec/div***

Header: TB:ScaleB

Type: Float

Argument: 10e-12 to 50e-3

Action: set scale of delayed timebase when time units are used

### ***Main timebase scale, bit/div***

Header: TB:BitScaleA

Type: Float

Argument: depends on actual bit rate

Action: set scale of main timebase when bit units are used

### ***Delayed timebase scale, bit/div***

Header: TB:BitScaleB

Type: Float

Argument: depends on actual bit rate

Action: set scale of delayed timebase when bit units are used

**Timebase delay**

Header: TB:Delay

Type: Float

Argument: 0 to 10

Action: set delay of intensified, delayed timebase in divisions

**Dual delayed timebase**

Header: TB:DualDel

Type: On/off

Action: turn on or off the dual delayed time base (used in intensified or delayed timebase)

**Timebase delta delay**

Header: TB:DeltaDel

Type: Float

Argument: 0 to 10

Action: set delta delay of intensified, delayed timebase in divisions (used in dual delayed timebase)

## 4.6 Trigger commands

**Trigger Source**

Header: Trig:Source

Type: Selector

Arguments: Direct, ExtHF, IntClock, ClockRecov

Action: set trigger source

**Direct Input Trigger Level**

Header: Trig:ExtLevel

Type: Float

Argument: -1 to +1

Action: set trigger level for direct input, volts

**Trigger Period for Internal Clock Sources**

Header: Trig:IntRate

Type: Float

Argument: 16e-9 to 0.002

Action: set period for internal clock trigger source, seconds

### **Trigger Mode**

Header: Trig:Mode

Type: Selector

Arguments: Free, Trig

Action: set Freerun or Triggered mode of the trigger

### **Direct Trigger Slope**

Header: Trig:Slope

Type: Selector

Arguments: Pos, Neg

Action: set Positive or Negative slope of trigger

### **Holdoff Time**

Header: Trig:Holdoff

Type: Float

Argument: 5e-6 to 1

Action: set the holdoff time, seconds

### **Direct Trigger Hysteresis**

Header: Trig:Hister

Type: Selector

Arguments: Norm, HighSens

Action: set hysteresis for direct trigger (Norm) or set off (HighSens)

### **Attenuator Unit for Direct Input**

Header: Trig:Atten:Unit

Type: Selector

Arguments: Off, Ratio, DB

Action: set presence and unit of attenuator or converter used with direct trigger input

### **Direct Input Attenuation (ratio)**

Header: Trig:Atten:Ratio

Type: Float  
 Argument: 0.0001 to 1000000  
 Action: set attenuation ratio. This setting is active only when attenuator unit is ratio.

#### **Direct Input Attenuation (dB)**

Header: Trig:Atten:DB  
 Type: Float  
 Argument: -80 to +120  
 Action: set the attenuation in dB. This setting is active only when attenuator unit is decibels.

### 4.7 Acquisition commands

#### **Type of signal**

Header: Acq:FitTo  
 Type: Selector  
 Arguments: Multi, Single  
 Action: prepare the instruments for best acquisition of single-valued or multi-valued

#### **Sampling Mode**

Header: Acq:Sampl  
 Type: Selector  
 Arguments: Simult, Altern  
 Action: Simult – set simultaneous acquisition on Channels 1 and 2  
 Alternate – set alternate acquisitions on Channels 1 and 2

#### **Acquisition Mode of Channel**

Header: Acq:Ch1:Mode; Acq:Ch2:Mode  
 Type: Selector  
 Arguments: Sample, AvgStab, AvgMult, EnvMinMax, EnvMax, EnvMin  
 Action: set acquisitions mode of specified channel

#### **Channel Averaging**

Header: Acq:Ch1:NAvg, Acq:Ch2:NAvg  
 Type: Integer

Argument: 1, 2, 4, 8, 16, ... 4096  
Action: set averaging coefficient for specified channel

### **Channel Envelopes**

Header: Acq:Ch1:NEnv, Acq:Ch2:NEnv  
Type: Integer  
Argument: 1, 2, 4, 8, 16, ... , 4096, 8192  
Action: set number of signals for envelope mode for specified channel.  
Argument 8192 is used for unlimited number of signals.

### **Channel Record Length**

Header: Acq:Ch1:RecLen, Acq:Ch2:RecLen  
Type: Integer  
Argument: 32, 64, 128, ... , 4096  
Action: set number of points for specified channel

### **Termination of Acquisition**

Header: Acq:RunUntil  
Type: Selector  
Arguments: StopBtn; NAcq  
Action: set condition for terminating acquisition – when the Stop Button pressed or after specified number of waveforms is reached

### **Number of Waveforms**

Header: Acq:NAcq  
Type: Integer  
Argument: 1 to 65535  
Action: set number of signals for terminating acquisition

### **Action when Number of Waveforms reached**

Header: Acq:React  
Type: On/off-group  
Items: Beep, Save  
Action: if Save is turned on, every signal is stored to disk;  
if Beep is turned on, the beep signal will sound after the specified number of waveforms is reached

## 4.8 Display commands

Mnemonic `<src>` in some Display Commands signifies Source  
 ( `<src>` is: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2)

### **Trace mode**

Header: `Displ:TraceMode`

Type: Selector

Arguments: `AllLocked, PerTrace`

Action: in `PerTrace` mode, every waveform may be displayed in its own style;  
 in `AllLocked` mode, the display style of all waveforms is set as the style of the active trace

### **Select active trace**

Header: `Displ:TraceSel`

Type: Selector

Arguments: `Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2, XY`

Action: select active trace for `AllLocked` trace mode

### **Set Display Style**

Header: `Displ:<src>:Style`

Type: Selector

Arguments: `Dots, Vectors, VarPersist, InfinPers, VarGrayScal, InfGrayScal, VColorGrad, IColorGrad`

Action: set display style for specified trace in `PerTrace` mode;  
 set display style for all traces in `AllLocked` mode if `<src>` equal to the active trace, or do nothing if `<src>` not equal to the active trace

### **Persistence Time , seconds (for VarPersist Style)**

Header: `Displ:<src>:PersistTime`

Type: Float

Argument: 0.1 to 20

Action: set persistence time for specified trace in `PerTrace` mode;  
 set persistence time for all traces in `AllLocked` mode if `<src>` is equal to the active trace, or do nothing if `<src>` not equal to the active trace

**Refresh Time, seconds (for VarGrayScal or VColorGrade Styles)**

Header: Displ:<src>:RefreshTime

Type: Float

Argument: 1 to 200

Action: set refresh time for specified trace in PerTrace mode;  
set refresh time for all traces in AllLocked mode if <src> equal to the active trace, or do nothing if <src> not equal to the active trace

**Reset Display Style**

Header: Displ:ResetAll

Type: Execution

Action: Reset Display Styles to initial state (variable persistence 2 c)

**Display Format**

Header: Displ:Format

Type: Selector

Arguments: YT, 2YT, 4YT, XY, CombYTTY, Comb2YTTY

Action: select number and kinds of screens

**Define Trace Screen (for 4YT Format)**

Header: Displ:Screen4:<trace>,  
where <trace> is Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2, Hist

Type: Selector

Arguments: 1, 2, 3, 4

Action: move specified trace onto specified screen in 4YT format

**Define Trace Screen (for 2YT, Comb2YTTY Formats)**

Header: Displ:Screen2:<trace>,  
where <trace> is Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2, Hist

Type: Selector

Arguments: 1, 2

Action: move specified trace onto specified screen in 2YT or Comb2YTTY formats

**Source of X Axis for XY Screen**

Header: Displ:XAxis

Type: Selector  
 Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2  
 Action: set specified signal as X axis for XY screen

**Source of Y Axis for XY Screen**

Header: Displ:YAxis  
 Type: Selector  
 Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2  
 Action: set specified signal as Y axis for XY screen

**Graticule Type**

Header: Displ:Gratic  
 Type: Selector  
 Arguments: Grid, Frame, Axis, Off  
 Action: define type of graticule for YT and XY screens

4.9 Save/Recall commands

4.9.1 Work with Memo Zones (M1, M2, M3, M4)

**Memory Display**

Header: Save:Memo:On  
 Type: On/off-group  
 Items: M1, M2, M3, M4  
 Action: control display of memory zones

**Source for storing into Memory**

Header: Save:Memo:Source  
 Type: Selector  
 Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2  
 Action: define signal as source for storing into memory zone

**Select Memory for Saving**

Header: Save:Memo:ToMemo  
 Type: Selector



Arguments: M1, M2, M3, M4

Action: define memory zone for saving

### **Save into Memory**

Header: Save:Memo:Save

Type: Execution

Action: store selected source into selected memory

## 4.9.2 Work with Disk

### **Source for saving to file**

Header: Save:Memo:Source

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: define signal as source for saving to file

### **File Name**

Header: Save:Disk:FileName

Type: Data

Argument: text string

Forms: command, query, command with query

Action: define file name for saving specified signal to disk

### **File Name Mode**

Header: Save:Disk:NameMode

Type: Selector

Arguments: Manual, Auto

Action: set file name mode. In `Auto` mode the file name consists of a base name followed by an underscore (`_`) and a five-digit number. Each time you save a waveform, the number in the file name is automatically incremented. For example: `basename_00001.wfm`, `basename_00002.wfm`, `basename_00003.wfm` and so on.

### **Format of stored files**

Header: Save:Disk:FileFormat

Type: Selector

Arguments: Binary, Verbose, YOnly

Action: set file format

**Save to Disk**

Header: Save:Disk:Save

Type: Execution

Action: save selected source to previously specified file

**Select Memory for loading signal from disk**

Header: Save:Disk:ToMemo

Type: Selector

Arguments: M1, M2, M3, M4

Action: selects which of the available memory locations the instrument loads the saved file into

**Load**

Header: Save:Disk:Load

Type: Execution

Action: load the previously specified disk file into the previously specified Memory Zone

4.9.3 Work with Setups

**Recall Factory Setup**

Header: Save:Setup:RecFact

Type: Execution

Action: returns instrument to manufacturer's default setting

**Recall Power-Off Setup**

Header: Save:Setup:RecLast

Type: Execution

Action: returns instrument to last setting before power supply was last switched off

**Save Setup as Default**

Header: Save:Setup:SvAsDefault

Type: Execution

Action: stores present front-panel setup as default setup

**Name of Custom Setup File**

Header: Save:Setup:FileName  
 Type: Data  
 Argument: text string  
 Forms: command, query, command with query  
 Action: define file name for storing Custom Setup

**Save Custom Setup**

Header: Save:Setup:Save  
 Type: Execution  
 Action: stores present front-panel setup as previously specified custom setup

**Recall Custom Setup**

Header: Save:Setup:Recall  
 Type: Execution  
 Action: recall setup previously saved to file. The name of the setup must first be defined by the command Save:Setup:FileName.

## 4.10 Markers commands

**Marker Type**

Header: Mark:Type  
 Type: Selector  
 Arguments: Off, MX, MY, XY  
 Action: set marker type

**Marker Sources**

Header: Mark:M1:Source, Mark:M2:Source  
 Type: Selector  
 Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2  
 Action: attach specified marker to specified signal

**X position of Marker**

Header: Mark:M1:XPos, Mark:M2:XPos  
 Type: Float  
 Argument: real value of X-axis

Action: set the X position of specified marker

### **Y position of Marker**

Header: Mark:M1:YPos, Mark:M2:YPos

Type: Float

Argument: real value of Y-axis

Action: set Y position of specified marker

### **Motion of Markers**

Header: Mark:Motion

Type: Selector

Arguments: Independ, Paired

Action: when Paired motion is selected, you can move both markers with the M1 POSITION variable simultaneously, while the difference between markers can be moved with the M2 POSITION variable.

## 4.11 Measure commands

The mnemonic <src> in some Measure Commands signifies the Source ( <src> is: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2)

### 4.11.1 Measurements of Time Domain Signals

#### **Measurement Type**

Header: Meas:Display

Type: Selector

Arguments: Off, Param, Statistic

Action: set measurement type

#### **Measurement Source**

Header: Meas:DisplSrc

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: set source for Measurement

#### **Viewing of Define Parameters**

Header: Meas:View

Type: On/off

Action: set visibility of 'define parameters' markers for selected sources

### **Measurement Mode**

Header: Meas:Mode

Type: Selector

Arguments: Permanent, Single

Action: set measurement mode

### **Execute Single Measurement**

Header: Meas:SingleMeas

Type: Execution

Action: execute single measurement in Single mode

### **Statistic Measurement Mode**

Header: Meas:StatMode

Type: Selector

Arguments: Permanent, Window, Weight

Action: set Statistic Measurement mode

### **Windows Value**

Header: Meas:Window

Type: Integer

Argument: 8 to 8192

Action: set number of recently acquired waveforms for Window mode of Statistic Measurement

### **Weight Value**

Header: Meas:Weight

Type: Integer

Argument: 8 to 8192

Action: set weight variable for Weight mode of Statistic Measurement

### **Top/Base Definition Method**

Header: Meas:<src>:Method

Type: Selector

Arguments: Hist, MinMax, Marker

Action: sets Top and Base vertical reference thresholds for amplitude measurements of specified signals

#### **Top Value for Marker Method**

Header: Meas:<src>:Top

Type: Integer

Argument: 2 to 1023

Action: sets Top vertical reference threshold for specified signals. Argument 0 corresponds to the bottom of the screen, and argument 1023 corresponds to the top of the screen independently of the real screen's height

#### **Base Value (for Marker Method)**

Header: Meas:<src>:Base

Type: Integer

Argument: 1 to 1022

Action: sets Base vertical reference threshold for specified signals. Argument 0 corresponds to the bottom of the screen, and argument 1023 corresponds to the top of the screen independent of the real screen's height.

#### **Threshold Definition Method**

Header: Meas:<src>:Thresh

Type: Selector

Arguments: 10-90, 20-80, Custom

Action: sets lower, middle, and upper thresholds for measurements of the specified signals. May be set to the fixed values 10%-50%-90%; 20%-50%-80%; or custom values.

#### **Threshold Units**

Header: Meas:<src>:Unit

Type: Selector

Arguments: Percent, Volt, Division

Action: sets units of thresholds for specified signals. It used for custom threshold definition method only.

#### **Position of Upper, Middle or Lower Threshold**

Headers:

Meas:<src>:UpThresh

Meas:<src>:MidThresh

Meas:<src>:LowThresh

Type: Float

Arguments:

absolute voltage value(for Volt threshold units only)

-4 to +4 (for Division threshold units only)

Action: sets the threshold position for the specified signals

### **Percentage of Upper, Middle or Lower Threshold**

Headers:

Meas:<src>:UpThPerc

Meas:<src>:MidThPerc

Meas:<src>:LowThPerc

Type: Integer

Arguments: -80 to +200

Action: sets the threshold percentage for the specified signals. It used for Percent threshold units only. Argument 0 (%) correspond to the Base of the signals, and argument 100 (%) corresponds to the Top of the signals.

### **Margins Definition Mode**

Header: Meas:<src>:MargMode

Type: Selector

Arguments: Slope, Marker

Action: sets margins definition mode

### **Slope of Left or Right Margins**

Headers:

Meas:<src>:LeftSlope

Meas:<src>:RightSlope

Type: Integer

Arguments: 0 to 127

Action: sets the margin for the specified signals on the specified slope. It used for slope margins definition mode only. Argument 0 means the first rise, value 1 is first fall, 2 – second rise; 3 – second fall, and so on.

**Thresholds of Left and Right Margin Slopes**

Headers:

Meas:<src>:LeftTresh

Meas:<src>:RightTresh

Type: Selector

Arguments: Upper, Middle, Lower

Action: sets the thresholds for definitions of the left or right slope. It used for slope margins definition mode only.

**Position of Left or Right Margin**

Headers:

Meas:<src>:LeftMarker

Meas:<src>:RightMarker

Type: Float

Arguments: absolute time value

Action: sets the position of margin for the specified signals. It used for marker margins definition mode only.

**List of X Measurements**

Header: Meas:<src>:XParam

Type: On/off-group

Items: Period, Freq, PosWidth, NegWidth, Rise, Fall, PosDuty, NegDuty, PosCross, NegCross, BurstWidth, Cycles, TimeOfMax, TimeOfMin, PosJitterPp, PosJitterRMS, NegJitterPp, NegJitterRMS

Action: define the set of the X-axis measurements for the specified signals

**List of Y Measurements**

Header: Meas:<src>:YParam

Type: On/off-group

Items: Max, Min, PP, Top, Base, Ampl, Middle, Mean, dcRMS, acRMS, Area, CycMean, CycDcRMS, CycAcRMS, CycArea, PosOver, NegOver

Action: define the set of the Y-axis measurements for the specified signals

**Second Source for Inter-Signal Measurements**

Header: Meas:Source2



Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: set the second source for the inter-signal measurements

#### **List of X Inter-Signal Measurements**

Header: Meas:<src>:XDualPar

Type: On/off-group

Items: DellR1R, DellR1F, DellF1R, DellF1F, DellRnR, DellRnF, DellFnR, DellFnF, PhaseDeg, PhaseRad, PhasePerc

Action: define the set of the X-axis inter-signal measurements for the specified signals

#### **List of Y Inter-Signal Measurements**

Header: Meas:<src>:YDualPar

Type: On/off-group

Items: Gain, DBGain

Action: define the set of the Y-axis inter-signal measurements for the specified signals

#### **Delete all Measures for all Sources**

Header: Meas:ClearAll

Type: Execution

Action: Clear list of all measurements for all signals

### 4.11.2 Measurements of Spectrum Signals

#### **Limits Definition Method for Spectrum**

Header: Meas:<src>:FFTMethod

Type: Selector

Arguments: Harmonic, Peak

Action: sets the method of the limits definition for the specified signal. It used for spectrum signals only.

#### **Left and Right Spectrums Margin**

Headers:

Meas:<src>:FFTLeft

Meas:<src>:FFTRight

Type: Float

Arguments: absolute frequency value

Action: sets the position of margin for the specified spectrum signals. It used for searching for peak 1 of the spectrum for the Harmonic method.

### **Peak Level of Spectrum**

Header: Meas:<src>:PeakLevel

Type: Float

Arguments: -100 to +80 (dBV)

Action: sets the level for the specified spectrum signals. It used for searching a peak of the spectrum for the Peak method.

### **Left and Right Spectrum Peaks**

Headers:

Meas:<src>:PeakLeft

Meas:<src>:PeakRight

Type: Integer

Arguments: 1 to 41

Action: sets the first and second peaks for the specified spectrum signals

### **List of Spectrum Frequency Measurements**

Header: Meas:<src>:XFFTPar

Type: On/off-group

Items: Freq, DFreq

Action: define the set of the frequency measurements for the specified signals

### **List of Spectrum Magnitude Measurements**

Header: Meas:<src>:YFFTPar

Type: On/off-group

Items: Magn, DMagn, TDH

Action: define the set of the magnitude measurements for the specified signals

## 4.11.3 Getting Measurement Results

**Get List of Measured Parameters**

|           |                                                                                         |
|-----------|-----------------------------------------------------------------------------------------|
| Header:   | Meas:Res:List?                                                                          |
| Type:     | Data                                                                                    |
| Argument: | none                                                                                    |
| Forms:    | query only                                                                              |
| Action:   | return text with the list of the active measurements for all signals with ordinal index |

**Get Current Value of Parameter**

|                |                                                            |
|----------------|------------------------------------------------------------|
| Header:        | Meas:Res:<N>?                                              |
| Parameter <N>: | index of the parameter in the list                         |
| Type:          | Data                                                       |
| Argument:      | none                                                       |
| Forms:         | query only                                                 |
| Action:        | return the last result of the specified measured parameter |

**Get Statistic Value of Parameter**

|                  |                                                                    |
|------------------|--------------------------------------------------------------------|
| Header:          | Meas:Res:<N>:<Val>?                                                |
| Parameter <N>:   | index of the parameter in the list                                 |
| Parameter <Val>: | Wfm, Min, Max, Mean, StdDev                                        |
| Type:            | Data                                                               |
| Arguments:       | none                                                               |
| Forms:           | command with query only                                            |
| Action:          | return the specified statistic parameter of the measured parameter |

## 4.12 Limit Tests commands

**Limit Test On/Off**

|         |                                                                                                         |
|---------|---------------------------------------------------------------------------------------------------------|
| Header: | Limit:TestOn                                                                                            |
| Type:   | On/off                                                                                                  |
| Action: | Enable/disable the Limit Test. Must be set On after full definition of all other Limit Test parameters. |

**Limit Test Termination Condition**

Header:       Limit:RunUntil  
 Type:         Selector  
 Arguments:   StopBtn, Failur, Wfm  
 Action:       set condition of Limit Test Termination

**Number of Failures**

Header:       Limit:Failures  
 Type:         Integer  
 Argument:     1 to 10000  
 Action:       set number of failures for the Failur Condition of the Limit

**Number of Waveforms**

Header:       Limit:NWfms  
 Type:         Integer  
 Argument:     1 to 1000000  
 Action:       set the number of waveforms for the Ffm Condition of the Limit

**Action**

Header:       Limit:Action  
 Type:         On/off-group  
 Items:        Beep, Save, Stop  
 Action:       Save: every signal with a limit condition is stored to the disk;  
               Beep: the beep signal will sound for every limit condition;  
               Stop: acquisition immediately stops after the first limit  
               condition

**Action If**

Header:       Limit:If  
 Type:         Selector  
 Arguments:   AnyFail, AllPass, AllFail, AnyPass  
 Action:       define the limit condition:  
               AnyFail - one or more active measures fails;  
               AllPass - all active measures are good;  
               AllFail - all active measures fail;  
               AnyPass - one or more active measurements is good

**Format of Stored Files**

Header: Limit:FileFormat  
Type: Selector  
Arguments: Binary, Verbose, YOnly  
Action: set file format

**File Name**

Header: Limit:FileName  
Type: Data  
Argument: text string  
Forms: command, query, command with query  
Action: define file name for saving the specified signals to disk

**Parameter Activity**

Headers:  
Limit1:Activ  
Limit2:Activ  
Limit3:Activ  
Limit4:Activ  
Type: On/off  
Action: Enable/disable the Limit Test for relevant parameter

**Parameter Limit Mode**

Headers  
Limit1:Mode  
Limit2:Mode  
Limit3:Mode  
Limit4:Mode  
Type: Selector  
Arguments: Center, Limit  
Action: set mode of limits for the relevant parameter

**Upper and Lower Limits of Parameters**

Headers:

```

Limit1:UpLimit Limit1:LowLimit
Limit2:UpLimit Limit2:LowLimit
Limit3:UpLimit Limit3:LowLimit
Limit4:UpLimit Limit4:LowLimit

```

Type: Float

Arguments: absolute value of limit

Action: sets the limit's value. It is used only for `Limit` mode of the parameter's limit.

**Parameter Center Mode**

Headers

```

Limit1:CenterMode
Limit2:CenterMode
Limit3:CenterMode
Limit4:CenterMode

```

Type: Selector

Arguments: `CurrMean`, `UserDef`

Action: set the mode of the center definition for the relevant parameter. It used only for the `Center` mode of the parameter limit.

**Center Value**

Headers:

```

Limit1:CenterVal
Limit2:CenterVal
Limit3:CenterVal
Limit4:CenterVal

```

Type: Float

Arguments: absolute value of center

Action: set the absolute center value. It used for `UserDef` mode of the center of the parameter.

**Parameter Delta Mode**

Headers:

Limit1:Delta

Limit2:Delta

Limit3:Delta

Limit4:Delta

Type: Selector

Arguments: StdDev, UserDef, UserPerc

Action: set mode of delta definition for relevant parameter. It used for Center mode of parameter limit only.

**Parameter Delta Value for Standard Deviation mode**

Headers:

Limit1:StdDev

Limit2:StdDev

Limit3:StdDev

Limit4:StdDev

Type: Float

Arguments: 0.1 to 100 standard deviations of the parameter

Action: sets the delta value. It used for StdDev mode of parameter delta only.

**Parameter Delta Value for User Defined Mode**

Headers:

Limit1&gt;UserDef

Limit2&gt;UserDef

Limit3&gt;UserDef

Limit4&gt;UserDef

Type: Float

Arguments: absolute value of delta

Action: sets the delta value. It used for UserDef mode of delta of the parameter only.

**Parameter Delta Percentage for User Defined mode**

Headers:

Limit1:UserPerc

Limit2:UserPerc

Limit3:UserPerc

Limit4:UserPerc

Type: Float

Arguments: 0.01% to 90% standard deviations of the parameter

Action: sets the delta value. It used for UserPerc mode of delta of the parameter only.

**Failure When**

Headers:

Limit1:FailWhen

Limit2:FailWhen

Limit3:FailWhen

Limit4:FailWhen

Type: Selector

Arguments: Outside, Inside, Always

Action: set the mode of the quality control for the according parameter

**If Measurement Undefined**

Headers:

Limit1:NotFound

Limit2:NotFound

Limit3:NotFound

Limit4:NotFound

Type: Selector

Arguments: Ignore, Fail, Pass

Action: set limit status when measurement is undefined



## 4.13 Mathematics commands

### **Enable Mathematical Function**

Headers:

F1:Display

F2:Display

F3:Display

F4:Display

Type: On/off

Action: enable/disable the calculation and display of the relevant functions

### **Function Operator**

Headers:

F1:Operat

F2:Operat

F3:Operat

F4:Operat

Type: Selector

Arguments: Add, Sub, Mult, Div, Invert, Abs, Exp\_e, Exp\_10, Log\_e, Log\_10, Dif\_al, Int\_al, IFFT, LinInt, SinInt, Smooth, Trend

Action: set the operator of the specified function

### **Operand 1**

Headers:

F1:Source1

F2:Source1

F3:Source1

F3:Source1

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: set the first operand of the specified function

### **Operand 2**

Headers:

F1:Source2

F2:Source2

F3:Source2

F4:Source2

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2, Const

Action: set the second operand of the specified function. Used for Add, Sub, Mult, Div operators only.

### **Constant Value**

Headers:

F1:Const

F2:Const

F3:Const

F4:Const

Type: Float

Arguments: absolute value of constant

Action: set the constant for the specified function. Used when Const is the second operand.

### **Smoothing Parameter**

Headers:

F1:SmoothLen

F2:SmoothLen

F3:SmoothLen

F4:SmoothLen

Type: Integer

Argument: 3, 5, 7, 9, ... , 49, 51

Action: set the length of the smoothing interval in points for specified function. Used for Smooth operator only.

### **Trend Measurement**

Headers:

F1:TrendMeas

F2:TrendMeas

F3:TrendMeas

F4:TrendMeas

Type: Selector

Arguments: Period, Freq, PosWidth, NegWidth, RiseTime, FallTime, PosDuty, NegDuty

Action: set the kind of trend for the specified function. Used for Trend operator only.

#### 4.14 FFT commands

##### **Enable Spectra**

Headers:

Spectr1:Display

Spectr2:Display

Type: On/off

Action: enable/disable the calculation and display of the relevant spectrum

##### **Spectrum Source**

Headers:

Spectr1:Source1

Spectr2:Source1

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4

Action: set the source of the specified spectrum

##### **Window**

Headers:

Spectr1:Window

Spectr2:Window

Spectr3:Window

Spectr4:Window

Type: Selector  
 Arguments: Rectang, Hamming, Hanning, Flattop, BlackHarr, KaiserBess  
 Action: set the window for specified spectrum

## 4.15 Histogram commands

### 4.15.1 Setting Histogram Parameters

#### **Histogram Axis**

Header: Hist:Axis  
 Type: Selector  
 Arguments: Off, Vert, Horiz  
 Action: set axis of histogram

#### **Histogram Source**

Header: Hist:Source  
 Type: Selector  
 Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2  
 Action: select specified signal as source of histogram

#### **Histogram Visibility**

Header: Hist:Visible  
 Type: On/off  
 Action: set visibility of histogram. Acquisition of the histogram proceeds independently of this command.

#### **Histogram Finish Condition**

Header: Hist:RunUntil  
 Type: Selector  
 Arguments: StopSingle, Wfms, Samples  
 Action: set finish condition for acquiring histogram

#### **Number of Waveforms for Histogram**

Header: Hist:NWfm  
 Type: Integer  
 Argument: 1 to 1000000

Action: set number of signals for termination of histogram acquisition

### **Number of Samples for Histogram**

Header: `Hist:NSample`

Type: Integer

Argument: 1 to 10000000

Action: set number of samples for termination of histogram acquisition

### **Limit Mode for Histogram Window**

Header: `Hist:Limits`

Type: Selector

Arguments: `Paried`, `Independ`

Action: set the mode of the limits of histogram window

### **Limit Units for Histogram Window**

Header: `Hist:Units`

Type: Selector

Arguments: `Absolute`, `Percent`

Action: set the units of the limits of histogram window

### **Left and Right Window Limits for Vertical or Horizontal Histogram**

Headers: `Hist:WVert:Left` `Hist:WVert:Right`

`Hist:WHor:Left` `Hist:WHor:Right`

Type: Float

Argument: real value of the X-axis (for `Absolute` units)

0% to 100% of the X-axis (for `Percent` units)

Action: set the X positions of the histogram window

### **Top and Bottom Window Limits for Vertical or Horizontal Histogram**

Headers: `Hist:WVert:Top` `Hist:WVert:Bottom`

`Hist:WHor:Top` `Hist:WHor:Bottom`

Type: Float

Argument: real value of the Y-axis (for `Absolute` units)

0% to 100% of the Y-axis (for `Percent` units)

Action: set the Y positions of the histogram window

### **Window Visibility**

Header: Hist:Display

Type: On/off

Action: set visibility of the window

### **Calculation Mode**

Header: Hist:Mode

Type: Selector

Arguments: Normal, Exponent

Action: set mode of histogram calculation

### **Weight for Exponential Calculation**

Header: Hist:Weight

Type: Integer

Argument: 8, 16, 32, ..., 8192

Action: set the number of signals for termination of acquisition

### **Scale Type**

Header: Hist:ScaleType

Type: Selector

Arguments: Linear, Logarith

Action: set the type of the histogram scale

### **Scale Mode**

Header: Hist:ScaleMode

Type: Selector

Arguments: Auto, Manual

Action: set the mode of the histogram scale

### **Linear Scale of Vertical or Horizontal Histogram**

Headers: Hist:VertScale

Hist:HorScale

Type: Float

Argument: (10 to 100) %/div

Action: set the scale of the histogram. Used for Manual mode and Linear type of scale only.

#### ***Linear Offset of Vertical or Horizontal Histogram***

Headers: Hist:VertOffset

Hist:HorOffset

Type: Float

Argument: 0% to 100%

Action: set the offset of the histograms. It used for Manual mode and Linear type of scale only.

#### ***Logarithmic Scale of Vertical or Horizontal Histogram***

Headers: Hist:VertDBScale

Hist:HorDBScale

Type: Float

Argument: (6 to 60) dB/div

Action: set the scale of the histograms. Used for Manual mode and Logarithmic type of scale only.

#### ***Logarithmic Offset of Vertical or Horizontal Histogram***

Headers: Hist:VertDBOffs

Hist:HorDBOffs

Type: Float

Argument: (-60 to 0) dB

Action: set the offset of the histograms. Used for Manual mode and Logarithmic type of scale only.

## 4.15.2 Getting Results of Histogram

**Get Histogram Data**

Headers: `Hist:Data:Vert?`

`Hist:Data:Hor?`

Type: `Data`

Argument: `none`

Forms: `query only`

Action: `return text string with values of the histogram (comma-separated)`

**Get Histogram Measurement**

Headers: `Hist:Res:<Param>?`

Parameter `<Param>`:

- `InBox` - Number of Hints in Box
- `Wfm` - Number of Waveforms
- `Peak` - Peak Value of Histogram
- `PP` - Difference between highest and lowest Values of Signal
- `Median` - Centre between highest and lowest Values of Signal
- `Mean` - Average of Distribution of Histogram
- `StdDev` - Standard Deviation of Histogram
- `Mean1S` - number of hints in  $\text{Mean} \pm \text{StdDev}$  Region, %
- `Mean2S` - number of hints in  $\text{Mean} \pm 2\text{StdDev}$  Region, %
- `Mean3S` - number of hints in  $\text{Mean} \pm 3\text{StdDev}$  Region, %
- `Min` - Min. Value of Signal
- `Max` - Max. Value of Signal
- `Max-Max` - Difference between two maxima of histogram

Type: `Data`

Argument: `none`

Forms: `query only`

Action: `return text string with value of the specified parameters`



## 4.16 Mask Test commands

### 4.16.1 Common Mask Commands

#### **Mask Erasing**

Header: Mask:EraseMask

Type: Execution

Action: Clear the current mask from the display

#### **Signal for Mask Testing**

Header: Mask:CompareWith

Type: Selector

Arguments: Ch1, Ch2

Action: select the signal for mask testing

#### **Actuate Mask Testing**

Header: Mask:Test

Type: On/off

Action: enable/disable mask test execution

#### **User Masks File Name**

Header: Mask:MaskFile

Type: Data

Argument: text string

Forms: command, query, command with query.

Action: define the file name for next loading of the user mask from the disk

#### **Load User Mask**

Header: Mask:LoadUser

Type: Execution

Action: load the previously specified user mask

#### **Mask Test Finish Condition**

Header: Mask:RunUntil

Type: Selector

Arguments: StopBtn, FailedWfms, FailedSmpIs, Wfms, Samples

Action: set condition of Mask Test Termination

**Number of Failed Waveforms**

Header: Mask:FailWfms  
Type: Integer  
Argument: 1 to 1000000  
Action: set the number of the failed waveforms for the FailedWfms finish condition

**Number of Failed Samples**

Header: Mask:FailSmp1s  
Type: Integer  
Argument: 1 to 1000000  
Action: set the number of failed samples for the FailedSmp1s finish condition

**Number of Waveforms**

Header: Mask:NWfms  
Type: Integer  
Argument: 1 to 1000000  
Action: set number of waveforms for the Wfms finish condition

**Number of Samples**

Header: Mask:NSamples  
Type: Integer  
Argument: 1 to 1000000  
Action: set number of samples for the Samples finish condition

**Mask Test Actions**

Header: Mask:Action  
Type: On/off-group  
Items: Beep, Save  
Action: Save: every failed signal is stored to disk  
Beep: the beep signal will sound for every failed signal

**Format of Stored Files**

Header: Mask:FileFormat  
Type: Selector

Arguments: Binary, Verbose, YOnly

Action: set file format. Used when Save action is on.

### **Stored File Name**

Header: Mask:FileName

Type: Data

Argument: text string

Forms: command, query, command with query

Action: define file name for storing failed signals on Disk. Used when Save action is on.

## 4.16.2 Standard Mask Commands

### **Alignment of Signal with Standard Mask**

Header: StdMask:Align

Type: On/off

Action: enable/disable alignment of the tested signal with the standard mask parameters

### **Enable Margins**

Header: StdMask:MarginsOn

Type: On/off

Action: enable/disable the margin control of eye-type masks

### **Margins Value**

Header: StdMask:MarginsVal

Type: Float

Arguments: -100% to +100%

Action: set the margins value. Used when margins is enabled.

### **Get List of Standards**

Header: StdMask:StdsList?

Type: Data

Argument: none

Forms: query only

Action: return list of mask standards with ordinal index

**Select Standard**

Header: StdMask:StdIndex  
 Type: Integer  
 Argument: 0 to (number of standards-1)  
 Action: select the current standard by its ordinal index

**Get List of Masks**

Header: StdMask:MasksList?  
 Type: Data  
 Argument: none  
 Forms: query only  
 Action: return list of masks with ordinal index from the selected standard

**Select Standard Mask**

Header: StdMask:MaskIndex  
 Type: Integer  
 Argument: 0 to (number of masks in the current standard-1)  
 Action: load the specified mask by its ordinal index

## 4.16.3 Getting Mask Results

**Get Integrated Results of Mask Test**

Headers: Mask:Res:<Param>?

Parameter <Param>:

- AllWfm - number of waveforms
- FailWfm - number of failed waveforms
- AllSmpl - number of samples
- FailSmpl - number of failed samples

Type: Data  
 Argument: none  
 Forms: query only  
 Action: return text string with value of the specified parameter

**Get Number of Samples in Selected Polygons**

Headers: Mask:Res:Poly<N>?

Parameter <N>: number of the polygon, 1 to 8

Type: Data

Argument: none

Forms: query only

Action: return text string with value of failed samples on specified polygon

***Get Number of Samples in Margins of Selected Polygon***

Headers: Mask:Res:Poly<N>Mar?

Parameter <N>: number of the polygon, 1 to 4

Type: Data

Argument: none

Forms: query only

Action: return text string with value of failed samples on margin of specified polygon. Used when Margins enabled.

***Get Number of Samples in Selected Polygon with Margins Together***

Headers: Mask:Res:Poly<N>All?

Parameter <N>: number of the polygon, 1 to 4

Type: Data

Argument: none

Forms: query only

Action: return text string with total number of failed samples on the margin and on the specified polygon. Used when Margins enabled.

## 4.17 Eye Diagram commands

### 4.17.1 Setting Eye Parameters

***Type of Eye Measurements***

Header: Eye:Measure

Type: Selector

Arguments: Off, NRZ, RZ

Action: set type of eye measurements

***Source for Eye Measurements***

Header: Eye:Source

Type: Selector  
Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: set source for eye measurements

### **Number of Waveforms in One Measurement**

Header: Eye:WfmsInCycle

Type: Integer

Argument: 64, 128, 256, 512, 1024

Action: set number of waveforms in one measurement

### **Eye Frame Visibility**

Header: Eye:DispaylWind

Type: On/off

Action: set visibility of eye frame

### **Measurement Statistics**

Header: Eye:Statistic

Type: On/off

Action: enable/disable measurement statistics

### **Measurement Statistics Mode**

Header: Eye:Mode

Type: Selector

Arguments: Permanent, Window, Weight

Action: set mode of statistics calculation. Used when statistics enabled.

### **Window Value**

Header: Eye:Window

Type: Integer

Argument: 8, 16, 32, ..., 8192

Action: set window value. Used for Window mode of statistics.

### **Weight Value**

Header: Eye:Weight

Type: Integer

Argument: 8, 16, 32, ..., 8192

Action: set weight value. Used for Weight mode of statistics.

### ***Left and Right Boundary for NRZ Top/Base Finding***

Headers: Eye:LeftBound

Eye:RightBound

Type: Float

Argument: 10% to 90% of the NRZ period

Action: set the zone of the period of the NRZ signal for the top/base calculation.

### ***Threshold Definition Mode***

Header: Eye:TreshMode

Type: Selector

Arguments: 10-90, 20-80, Custom

Action: set mode of threshold definition

### ***Upper and Lower Threshold***

Headers: Eye:UpTresh

Eye:LowTresh

Type: Float

Argument: 5% to 95% of amplitude

Action: set the thresholds for the slopes calculation. Used for Custom mode only.

### ***List of X-axis NRZ Measurements***

Header: Eye:XNRZParam

Type: On/off-group

Items: Area, BitRate, BitTime, CrossTime, CycleArea, DutCycDistP, DutCycDistS, EyeWidth, EyeWidthP, FallTime, Freq, JitterPP, JitterRMS, Period, RiseTime

Action: define the set of the X-axis measurements for NRZ signals.

### ***List of Y-axis NRZ Measurements***

Header: Eye:YNRZParam

Type: On/off-group

Items: AcRMS, AvgPower, AvgPWdBm, CrossPerc, CrossLevel, ExtRatioDB, ExtRatioP, ExtRatio, EyeAmpl, EyeHeight, EyeHeightDB, Max, Mean, Mid, Min, NegOver, PPNoiseOne, PPNoiseZero, RMSNoiseOne, RMSNoiseZero, OneLevel, PeakPeak, PosOver, RMS, SNRaio, SNRaioDB, ZeroLevel

Action: define the set of Y-axis measures for NRZ signals

#### **List of X-axis RZ Measurements**

Header: Eye:XRZParam

Type: On/off-group

Items: Area, BitRate, BitTime, CycleArea, EyeWidth, EyeWidthP, FallTime, JittPpFall, JittPpRise, JittRMSFall, JittRMSRise, NegCross, PosCross, PosDutyCyc, PulseSymm, PulseWidth, RiseTime

Action: define the set of X-axis measurements for RZ signals

#### **List of Y-axis RZ Measurements**

Header: Eye:YRZParam

Type: On/off-group

Items: AcRMS, AvgPower, AvgPWdBm, Contrast, ContrastBb, ContrastP, ExtRatioDB, ExtRatioP, ExtRatio, EyeAmpl, EyeHeight, EyeHeightDB, EyeOpenFact, Max, Mean, Mid, Min, PPNoiseOne, PPNoiseZero, RmsNoiseOne, RMSNoiseZero, OneLevel, PeakPeak, RMS, SignToNoise, ZeroLevel

Action: define the set of the Y-axis measures for RZ signals.

### 4.17.2 Getting Eye Measurement Results

#### **Get List of Measured Parameters**

Header: Eye:Res:List?

Type: Data

Argument: none

Forms: query only

Action: return list of active eye measurements with ordinal index

#### **Get Current Value of Parameter**

Header: Eye:Res:<N>?

Parameter <N>: index of parameter in the list

Type: Data



Argument: none  
 Forms: query only  
 Action: return the result of the specified measured parameter

**Get Statistic Value of Parameter**

Header: Eye:Res:<N>:<Val>?  
 Parameter <N>: index of the parameter in the list  
 Parameter <Val>: Wfm, Min, Max, Mean, StdDev  
 Type: Data  
 Arguments: none  
 Forms: command with query only  
 Action: return the specified statistical parameter of the measured parameter

4.18 Utilities commands

**Start Autocalibration of Channels**

Header: Flash:Calibr:AutocalCh  
 Type: Execution  
 Action: Start self-calibration of channels

**Start Autocalibration of Timebase**

Header: Flash:Calibr:AutocalTB  
 Type: Execution  
 Action: Start self-calibration of timebase

**Get the Autocalibration status query**

Header: Flash:Calibr:AutocalResult?  
 Type: Integer  
 Action: Command is ignored, query returns an integer:

- 0 - Autocalibration finished OK;
- 1 - Signal must be disconnected from Ch1 Input. Autocalibration of the Channels is aborted.
- 2 - Signal must be disconnected from Ch2 Input. Autocalibration of the Channels is aborted.
- 3 - Signal must be disconnected from Ch1 and Ch2 Inputs. Autocalibration of the Channels is aborted.

5 - Autocalibration failed.

Version: This query can be used with PicoScope 9000 SW v.2.3.2 or later.

### ***When to Begin Autocalibration***

Header: Util:CalibrWhen

Type: On/off-group

Items: PowerOn, Period, Temperat

Action: PowerOn: autocalibration begins on the next Power On;  
 Period: autocalibration begins periodically with the specified interval;  
 Temperat: autocalibration begins when deviation of temperature inside the instrument exceeds the specified value

Note for the PicoScope 9000 SW v.2.3.2 or later.

When the GUI is in Invisible or RemoteOnly state the autocalibration routine can not be performed spontaneously and independently on this command (see at the GUI command above).

### ***Autocalibration Period***

Header: Util:CalPeriod

Type: Float

Argument: 0.5 to 16 hours

Action: sets autocalibration period in hours

### ***Temperature Deviation***

Header: Util:TempChange

Type: Float

Argument: 0.5 to 10 °C

Action: sets temperature deviation for autocalibration

### ***Get the Temperature of the Instrument query***

Header: Calibr:Temperature?

Type: Float

Argument: none

Forms: query only

Action: return the temperature inside the device in degrees Celsius

Version: This command can be used with PicoScope 9000 SW v.2.4.1 or later

## 4.19 Waveforms commands

This group of commands is designed for receiving acquired waveforms from the oscilloscope.

### **Waveform Source**

Header: Wfm:Source

Type: Selector

Arguments: Ch1, Ch1B2, Ch2, Ch2B2, F1, F2, F3, F4, M1, M2, M3, M4, S1, S2

Action: set the signal to be received

### **Spectrum Format**

Header: Wfm:Complex

Type: Selector

Arguments: Mod, Ph, Re, Im

Action: select the received component of the complex signal. Used for spectrum data.

### **Get Waveform Data**

Header: Wfm:Data?

Type: Data

Argument: none

Forms: query only

Action: return text string with values of all points of the signal (comma-separated)

### **Get Number of Points in Waveform**

Header: Wfm:Preamb:Poin?

Type: Data

Argument: none

Forms: query only

Action: return number of points in signal

### **Get X-axis Step**

Header: Wfm:Preamb:XInc?

Type: Data

Argument: none

Forms: query only  
 Action: return the increment on the X-axis for one signal point

**Get X-axis Origin**

Header: Wfm:Preamb:XOrg?  
 Type: Data  
 Argument: none  
 Forms: query only  
 Action: return the X-axis value for the first signal point

**Get X-axis Unit**

Header: Wfm:Preamb:XU?  
 Type: Data  
 Argument: none  
 Forms: query only  
 Action: return the X-axis physical units

**Get Y-axis Unit**

Header: Wfm:Preamb:YU?  
 Type: Data  
 Argument: none  
 Forms: query only  
 Action: return the Y-axis physical units

#### 4.20 System commands

This group of commands is used to control devices in a multi-instrument system.

**Count of Instruments in the system query**

Header: Instr:List:Count?  
 Type: Integer  
 Argument: none  
 Forms: query only  
 Action: return the number of devices in the system  
 Version: This command can be used with PicoScope 9000 SW v.2.4.0 or later

**Get a description of device number N query**

Header: Instr:List:Unit:N?  
when N = 0..count of instruments-1

Type: Data

Argument: none

Forms: query only

Action: return the description of the instrument number N in the format:  
N;User\_Name[Sys\_Name];Interface;IP

where:

N - number of devices in the system

User\_Name - user name of the Instrument, optional

Sys\_Name - system name of the Instrument from the list "ABSENT, DEFAULT\_DEMO, DEFAULT\_LAN, DEFAULT\_USB, DEMO2, DEMO3, UNIT1, UNIT2, UNIT3, UNIT4, UNIT5, UNIT6, UNIT7, UNIT8"

Interface - type of the interface from the list: "NONE, DEMO, LAN, USB"

IP - IP-address, for LAN interface only

Version: This command can be used with PicoScope 9000 SW v.2.4.0 or later.

**Get the name of the current device query**

Header: Instr:Current:Get?

Type: Data

Argument: none

Forms: query only.

Action: return the name of the current instrument in the format:  
User\_Name[Sys\_Name]

when:

User\_Name - user name of the Instrument, optional

Sys\_Name - system name of the instrument from the list "ABSENT, DEFAULT\_DEMO, DEFAULT\_LAN, DEFAULT\_USB, DEMO2, DEMO3, UNIT1, UNIT2, UNIT3, UNIT4, UNIT5, UNIT6, UNIT7, UNIT8"

Version: This command can be used with PicoScope 9000 SW v.2.4.0 or later

**Switch to another device command with query**

Header: Instr:Current:Set? Argument

Type: Data

Argument: Specifier of the required instrument. May be one of three alternatives: a number of devices in the system; a user name or a system name of the instruments. Note that all space symbols (" ") in the names must be changed to the underline symbol ("\_") in the commands

Forms: command with query only

Action: The GUI changes to the given device. If the switch is successful the command returns "OK". If the new device is already used by another interface the command returns "BUSY". If a new device is not connected to PC the command returns "NOT DETECTED"

Version: This command can be used with PicoScope 9000 SW v.2.4.0 or later

**GUI Ready query**

Header: Instr:GUIReady?

Type: On/Off

Argument: none

Forms: query only

Action: return "OFF" when the GUI is not finished loading and is not ready; and return "ON" when GUI is ready.

**ATTENTION!** This command must be the first after the start of the COM object. It must be repeated until the response is "ON".

Version: This command can be used with PicoScope 9000 SW v.2.4.1 or later.

## 5 Programming Examples

Your PicoScope installation includes programming examples in the following languages and development environments:

- Delphi
- LabVIEW
- Visual Basic .NET

### 5.1 Delphi

The program:

```
PicoScopeDelphiClientExample.dproj
```

in the `Delphi_Client_Example/` subdirectory of the PicoScope9000 SDK materials demonstrates how to operate PicoScope 9000 Series PC Oscilloscopes. The file:

```
PicoScope9000_TLB.pas
```

is the description of the PicoScope9000.COMRC object. You must include this file in your own programs. Other files required for the example are:

```
MainClient.pas
MainClient.dfm
PicoScopeDelphiClientExample.dproj
PicoScopeDelphiClientExample.dsk
PicoScopeDelphiClientExample.identcache
PicoScopeDelphiClientExample.res
```

All these 8 files must be put into the same directory and compiled. This has been tested with Delphi 2009.

### 5.2 LabVIEW

The program:

```
PicoScope_Example.vi
Test_Get_Data.vi
```

in the `LabView_Client_Example/` subdirectory of the PicoScope9000 SDK materials demonstrates how to operate PicoScope 9000 Series PC Oscilloscopes.

This files must be put into the some directory and compiled. It has been tested with LabVIEW Base Development System 8.2.1.

### 5.3 Visual Basic .NET

The project is located in the PicoScope9000VBdotNETClient/ subdirectory of the PicoScope9000 SDK. The subdirectory is a standard VB.NET project directory with the project file:

```
PicoScope9000VBdotNETClient.vbproj
```

solution file:

```
PicoScope9000VBdotNETClient.sln
MainForm.* files
```

and three subdirectories:

```
bin/
"My Project/"
obj/
```

Assembly `obj/*/Interop.PicoScope9000.dll`, which is referenced in the source code as `PicoScope9000.COMRC`, is a bridge between .NET platform and Windows Component Object Model (COM). It has been created with Visual Basic IDE by executing the command **Project | Add Reference... | COM | PicoScope9000**.

File `MainForm.vb` contains the whole source code of the example. Other files were created automatically either by IDE itself or by the visual form editor.

This example has been created and tested with Microsoft Visual Basic 2008 Express Edition.



# Index

## A

- Acquisition commands
  - Acquisition Mode of Channel 16
  - Action when Number of Waveforms reached 16
  - Channel Averaging 16
  - Channel Envelopes 16
  - Channel Record Length 16
  - Number of Waveforms 16
  - Sampling Mode 16
  - Termination of Acquisition 16
  - Type of signal 16
- All-off mode 6

## C

- Case-insensitivity 4
- Channels commands
  - Acquire a Channel 11
  - Attenuator dB 11
  - Attenuator linear/log 11
  - Attenuator ratio 11
  - Attenuator unit 11
  - Bandwidth of Channel 11
  - Channels commands 11
  - Display a Channel 11
  - Offset a Channel 11
  - Scale a Channel 11
- Characteristics 6
- Command classification 6
- Command entry rules 4
- Command header 3
- Command messages 2
- Commands 3
  - list of 10
- Common Mask Commands
  - Actuate Mask Testing 45
  - Format of Stored Files 45
  - Load User Mask 45
  - Mask Erasing 45
  - Mask Test Actions 45
  - Mask Test Finish Condition 45
  - Number of Failed Samples 45
  - Number of Failed Waveforms 45
  - Number of Samples 45
  - Number of Waveforms 45
  - Signal for Mask Testing 45
  - Stored File Name 45

- User Masks File Name 45
- COMRC Object 1
- Concatenating 4

## D

- Data-type commands 9
- Delphi 59
- Display commands
  - Define Trace Screen (for 2YT, Comb2YTX Y Formats) 18
  - Define Trace Screen (for 4YT Format) 18
  - Display commands 18
  - Display Format 18
  - Graticule Type 18
  - Persistence Time, seconds (for VarPersist Style) 18
  - Refresh Time, seconds (for VarGrayScal or VColorGrade Styles) 18
  - Reset Display Style 18
  - Select active trace 18
  - Set Display Style 18
  - Source of X Axis for XY Screen 18
  - Source of Y Axis for XY Screen 18
  - Trace mode 18

## E

- ExecCommand Method 1
- Execution-type commands 6

## F

- FFT commands
  - Enable Spectra 39
  - Spectrum Source 39
  - Window 39
- Float-type commands 8

## G

- Getting Eye Measurement Results, commands for
  - Get Current Value of Parameter 52
  - Get List of Measured Parameters 52
  - Get Statistic Value of Parameter 52
- Getting Mask Results, commands for
  - Get Integrated Results of Mask Test 48
  - Get Number of Samples in Margins of Selected Polygon 48
  - Get Number of Samples in Selected Polygon with Margins Together 48
  - Get Number of Samples in Selected Polygons 48
- Getting Measurement Results, commands for

Getting Measurement Results, commands for  
 Get Current Value of Parameter 31  
 Get List of Measured Parameters 31  
 Get Statistic Value of Parameter 31  
 Getting Results of Histogram, commands for  
 Get Histogram Data 44  
 Get Histogram Measurement 44  
 Group-off mode 6  
 Group-on mode 6  
 Group-query mode 6  
 GUI Ready query 56

## H

Header commands  
 Header 10  
 Headers 3

## I

Integer-type commands 8

## L

LabView 59  
 Limit Tests commands  
 Action 31  
 Action If 31  
 Center Value 31  
 Failure When 31  
 File Name 31  
 Format of Stored Files 31  
 If Measurement Undefined 31  
 Limit Test On/Off 31  
 Limit Test Termination Condition 31  
 Number of Failures 31  
 Number of Waveforms 31  
 Parameter Activity 31  
 Parameter Center Mode 31  
 Parameter Delta Mode 31  
 Parameter Delta Percentage for User Defined mode 31  
 Parameter Delta Value for Standard Deviation mode 31  
 Parameter Delta Value for User Defined Mode 31  
 Parameter Limit Mode 31  
 Upper and Lower Limits of Parameters 31  
 Line feeds 4  
 List of commands 10

## M

Markers commands

Marker Sources 23  
 Marker Type 23  
 Motion of Markers 23  
 X position of Marker 23  
 Y position of Marker 23

### Mathematics commands

Constant Value 37  
 Enable Mathematical Function 37  
 Function Operator 37  
 Operand 1 37  
 Operand 2 37  
 Smoothing Parameter 37  
 Trend Measurement 37

### Measure commands 24

Measurement of Spectrum Signals, commands for  
 Left and Right Spectrum Peaks 29  
 Left and Right Spectrums Margin 29  
 Limits Definition Method for Spectrum 29  
 List of Spectrum Frequency Measurements 29  
 List of Spectrum Magnitude Measurements 29  
 Peak Level of Spectrum 29

### Measurements of Time Domain Signals commands

Base Value (for Marker Method) 24  
 Delete all Measures for all Sources 24  
 Execute Single Measurement 24  
 List of X Inter-Signal Measurements 24  
 List of X Measurements 24  
 List of Y Inter-Signal Measurements 24  
 List of Y Measurements 24  
 Margins Definition Mode 24  
 Measurement Mode 24  
 Measurement Source 24  
 Measurement Type 24  
 Percentage of Upper, Middle or Lower Threshold 24  
 Position of Left or Right Margin 24  
 Position of Upper, Middle or Lower Threshold 24

### Second Source for Inter-Signal Measurements

Slope of Left or Right Margins 24  
 Statistic Measurement Mode 24  
 Threshold Definition Method 24  
 Threshold Units 24  
 Thresholds of Left and Right Margin Slopes 24

Top Value for Marker Method 24  
 Top/Base Definition Method 24  
 Viewing of Define Parameters 24  
 Weight Value 24  
 Windows Value 24

Messages 2

Mnemonics 4

## O

- On/Off Group-type commands 6
- On/Off-type commands 6
- Overview 2

## P

- PicoScope9000 COM Server 1
- Programming Examples 59

## Q

- Queries 3
- Query commands 2

## S

- Selector-type commands 8
- Semicolon 4
- Set commands 2
- Setting Eye Parameters, commands for
  - Eye Frame Visibility 49
  - Left and Right Boundary for NRZ Top/Base Finding 49
  - List of X-axis NRZ Measurements 49
  - List of X-axis RZ Measurements 49
  - List of Y-axis NRZ Measurements 49
  - List of Y-axis RZ Measurements 49
  - Measurement Statistics 49
  - Measurement Statistics Mode 49
  - Number of Waveforms in One Measurement
    - Source for Eye Measurements 49
    - Threshold Definition Mode 49
    - Type of Eye Measurements 49
    - Upper and Lower Threshold 49
    - Weight Value 49
    - Window Value 49
- Setting Histogram Parameters commands
  - Calculation Mode 40
  - Histogram Axis 40
  - Histogram Finish Condition 40
  - Histogram Source 40
  - Histogram Visibility 40
  - Left and Right Window Limits for Vertical or Horizontal Histogram 40
  - Limit Mode for Histogram Window 40
  - Limit Units for Histogram Window 40
  - Linear Offset of Vertical or Horizontal Histogram 40
  - Linear Scale of Vertical or Horizontal Histogram

- Logarithmic Offset of Vertical or Horizontal Histogram 40
- Logarithmic Scale of Vertical or Horizontal Histogram 40
- Number of Samples for Histogram 40
- Number of Waveforms for Histogram 40
- Scale Mode 40
- Scale Type 40
- Top and Bottom Window Limits for Vertical or Horizontal Histogram 40
- Weight for Exponential Calculation 40
- Window Visibility 40

- Single-item mode 6
- Specifications 6
- Standard Mask Commands
  - Alignment of Signal with Standard Mask 47
  - Enable Margins 47
  - Get List of Masks 47
  - Get List of Standards 47
  - Margins Value 47
  - Select Standard 47
  - Select Standard Mask 47
- System commands 56
  - Clear Display 10
  - Recall Default Setup 10
  - Start Autoscaling 10
  - Start Cycle Acquisition 10
  - Start Single Acquisition / Stop Acquisition 10
  - System commands 10

## T

- Times commands
  - Acquire a Channel 13
  - Attenuator dB 13
  - Attenuator linear/log 13
  - Attenuator ratio 13
  - Attenuator unit 13
  - Bandwidth of Channel 13
  - Display a Channel 13
  - Offset a Channel 13
  - Scale a Channel 13
  - Timebase Commands 13
- Trigger Commands
  - Acquire a Channel 14
  - Attenuator dB 14
  - Attenuator linear/log 14
  - Attenuator ratio 14
  - Attenuator unit 14
  - Bandwidth of Channel 14
  - Display a Channel 14
  - Offset a Channel 14
  - Scale a Channel 14

## U

### Utilities commands

- Autocalibration Period 53
- Start Autocalibration of Channels 53
- Start Autocalibration of Timebase 53
- Temperature Deviation 53
- When to Begin Autocalibration 53

## V

- Visual Basic .NET 60

## W

### Waveforms commands

- Get Number of Points in Waveform 55
- Get Waveform Data 55
- Get X-axis Origin 55
- Get X-axis Step 55
- Get X-axis Unit 55
- Get Y-axis Unit 55
- Spectrum Format 55
- Waveform Source 55

- White space 4

### Work with Disk commands

- File Name 21
- File Name Mode 21
- Format of stored files 21
- Load 21
- Save to Disk 21
- Select Memory for loading signal from disk 21
- Source for saving to file 21

### Work with Memo Zones (M1, M2, M3, M4) commands

- Memory Display 20
- Save into Memory 20
- Select Memory for Saving 20
- Source for storing into Memory 20

### Work with Setups commands

- Name of Custom Setup File 22
- Recall Custom Setup 22
- Recall Factory Setup 22
- Recall Power-Off Setup 22
- Save Custom Setup 22
- Save Setup as Default 22





## Pico Technology

James House  
Colmworth Business Park  
ST. NEOTS  
Cambridgeshire  
PE19 8YP  
United Kingdom  
Tel: +44 (0) 1480 396 395  
Fax: +44 (0) 1480 396 296  
[www.picotech.com](http://www.picotech.com)

ps9000pg.en-4

8.12.10

© 2010 Pico Technology. All rights reserved.