pico Technology

PicoScope® 9400 Series

Sampler Extended Real Time Oscilloscope

Programmer's Guide

ps9400pg.en-3

Contents

1 PicoScope 9400 API Reference

PicoScope 9400 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 9400 GUI application.

1.1 PicoScope9400 COM Server

The COM server implementing the API is called **PicoSample 4**, and is implemented by the PicoScope 9400 GUI application (**PicoSample4.exe**). It is registered in the system during the setup process, and can be explicitly unregistered and registered again by executing **PicoSample4.exe** with the **/UnregServer** or **/RegServer** switches.

1.2 ExecCommand Method

The **COMRC** object contains only one method, **ExecCommand**. This 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 either a query or a command with a query

The syntax of the commands and queries and the full list of commands are described in the following pages.

1.3 COMRC Object

To implement the API the server exposes only one object, **COMRC**. This object supports automation, so it can be used by high-level languages like JavaScript (HTML pages) or VBA (Microsoft Word). Additionally, low-level languages like C are also supported. The string defining the system-wide name of the object and used for object creation is **PicoSample4.COMRC**.

2 Commands Syntax

2.1 Command and Query Structure

2.1.1 Overview

The PicoScope 9400 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 the addition of a question mark at the end. For example, the set command

```
ACQuire:NACQuire
```

has a query form

```
ACQuire:NACQuire?
```

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

Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. If the command is concatenated with other commands, the header must begin with a colon (:).
<Mnemonic>	This is the header of the sub-function. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates items from one another.
<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 one another.
<Comma>	A single comma is used between the 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.

Command message elements

2.1.3 Commands

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

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

A command header consists of one or more mnemonics arranged in a hierarchy 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, whereas

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.

Query	Header Off	Header On
Ch1:Scale?	200 mV/div	CH1:SCALE 200 mV/div
ACQ:NAvg?	16	ACQ:NAVG 16

Comparison of Header Off and Header On responses

2.2 Command Entry

2.2.1 Rules

The following rules apply when entering commands:

- A mnemonic can be followed by any letters for easier understanding of the program's text. For example, these commands are all equivalent:

  ```
  Ch1:ATTEN:DIMENS Volt

  Ch1:ATTENuator:DIMENSion Volt

  Ch1:ATTENblabla:DIMENSblabla Volt
  ```

  However, arguments must not be followed by additional characters.

- 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 will ignore commands consisting of any combination of white space characters and line feeds.

2.2.2 Concatenation

You can concatenate any combination of set commands and queries by using a semicolon (;). The instrument executes concatenated commands in the order received. The following rules apply when concatenating commands and queries:

- You can separate completely different headers with a semicolon (;), and by adding a leading colon (:) at the beginning of all commands except the first one. For example

  ```
  TRIGger:MODE FREE
  ACQuire:NAVG 10
  ```

  can be concatenated into the single command

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

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

  ```
  Zoom1:Ch1:VertFactor 10
  Zoom1:Ch1:VertPosition -1
  ```

  into a single command

  ```
  Zoom1:Ch1:VertFactor 10; VertPosition -1
  ```

  The longer version also works equally well:

  ```
  Zoom1:Ch1:VertFactor 10;:Zoom1:Ch1:VertPosition -1
  ```

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


```
Acq:Mode Average;NAvg?
```

This 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:

  Displ:TraceMode AllLocked; ACQ:NAVG 10
  (a colon is needed before **ACQuire**)

  Displ:TraceMode AllLocked;:Format YT
  (there is an extra colon before **FORMAT**. Use **Displ:TraceMode AllLocked;Format YT** instead.)

  Displ:Ch1:Persistence Simple;Ch1:PersistTime 2
  (The levels of these mnemonics are different. Either remove the second use of **Ch1:** or place **:Displ:** in front of **Ch1:**.)

3 Command Classification

Most commands belong to one of a few types. For example, execution-type commands tell the instrument to perform a specific action, 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

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, with no *query* form.

3.2 On/off-type commands

On/off type commands tell the instrument to 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 corresponding function.

All on/off type commands have a query form, which will 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 on or off independently. 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 for each command is specified in the list of commands.

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 item of a command without changing its other items. In this case the item's mnemonic is added to the end of the command after a colon (**:**). This must be followed by a space character and then one of the following arguments: **On**, **Off**, **0**, **1**. For example, this 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
Group-on 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, this command turns on the rise time and fall time measurements for Channel 1:

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

Group-off mode
Group-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, this command turns off the 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 only has a query form. For example:

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

The answer may include one or more items separated by a comma, or **ClearAll** if all items are turned off. For example, the answer **Freq,Period** means there are two items turned on.

3.4 Selector-type commands

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

```
Trig:Analog:Ch1:Slope
```

has these possible arguments:

```
Pos, Neg, BiSlope
```

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 for each command is specified in the list of commands.

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

```
Trig:Analog:Ch1:Slope? Pos
```

This command sets the Direct input as the trigger source and returns **POS**.

3.5 Integer-type commands

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

```
INSTR:TimeBase:RecLen 1000
```

sets the length of signals to 1000 points. The valid range and increment of each value is different and is described in the list of commands.

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

```
INSTR:TimeBase:RecLen? 24
```

returns **50**, since 50 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.

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**, where **V/div** is a dimension of the scale, and the prefix **m** is milli.

The commands

```
INSTR:TimeBase:ScaleT? 0.0000001

INSTR:TimeBase:ScaleT? 100e-9

INSTR:TimeBase:ScaleT? 0.1u

INSTR:TimeBase:ScaleT? 100p
```

are equal and set the Scale 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 data to the instrument or to receive data from the instrument, such as the array of points from an acquired signal, the result of a measurement, and so on.

Some data-type commands only have a query form, 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.

4 Full list of commands

4.1 Header command

Header: **Header**

Type: On/Off

Action: Enables/disables headers as part of the query response

4.2 GUI commands

Header: **Gui**

Type: Selector

Arguments: **RemoteLocal, RemoteOnly, Invisible**

Action: Sets the behavior of the GUI when it is controlled by the COM-object

GUI ready query

Header: **Instr:GuiReady?**

Type: On/off-type command

Argument: none

Forms: query only

Action: Returns **On** when the GUI has finished loading and is ready to receive commands. Must be used first at system startup.

GUI control command

Header: **Gui:Control**

Type: Selector-type command

Arguments: **RemoteLocal, RemoteOnly, Invisible**

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

GUI side menu

Header: **Gui:SideMenu:Left:Menu**
Gui:SideMenu:Right:Menu

Type: Selector-type command

Arguments: **Off, Ch, Acq, Trig, Displ, Save, Mark, Meas, Math, Hist, Eye, Mask, Util**

Action: Remove or Set the specified side menu panel.

GUI side menu page

Header: **Gui:SideMenu:Left:Page**
Gui:SideMenu:Right:Page

Type: Integer-type command

Argument: 1 to N, when N is count of pages in the current side menu

Action: Select the page in the specified side menu panel.

Note: This command makes sense for side menus with two or more pages.

GUI side menu signal

Header: **Gui:SideMenu:Left:Signal**
Gui:SideMenu:Right:Signal

Type: Integer-type command

Argument: 1 to N, when N is count of active signals (max 4)

Action: Select the signal in the specified side menu panel.

Note: This command makes sense for the Channels, Save/Recall and Math menu.

4.3 System commands

Clear Display

Header: ***ClrDispl**

Type: Execution

Action: Clears the display immediately

Running Control

Header: ***RunControl**

Type: Selector

Arguments: **Stop, Single, Run**

Action:
Run – Start a continuous acquisition
Single – Start a single acquisition
Stop – Immediately stop the acquisition

Response:
Run – the instrument is in the continuous acquisition state
Single – the instrument is in the single acquisition state
Stop – the instrument is stopped

Start Autoscaling

Header: **`*Autoscale`**

Type: Selector

Arguments: None

Action: Starts autoscaling of the instrument

Recall Default Setup

Header: **`*DefSetup`**

Type: Execution

Action: Restores the instrument to its default setup

Set Copy Mode and Copy to the Clipboard

Header: **`*Copy:<Mode>`**

where **`<Mode>`** is one of:

`FullScreen`	**`FullWindow`**
`ClientPart`	**`InvClientPart`**
`ScopeScreen`	**`InvScopeScr`**

Type: Executing-type command

Action: Sets the specified copy mode (All display, software window, client part of the software window, client part of the software window with colors inverted, software screen area or software screen area with color inversion) and copy specified onto the clipboard.

Copy to the Clipboard

Header: **`*Copy`**

Type: Execution

Action: Puts the image onto the clipboard, depending on the Copy Mode

Get Copy Mode query

Header: **`*Copy?`**

Argument: None

Forms: Query only

Action: Returns current Copy Mode. See Set Copy Mode and Copy to the Clipboard.

4.4 Channel commands

General remark on channel commands
Some channel commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics Ch3 and Ch4 are not used for these models.

Display a Channel

Header: **`Ch1:Display`** **`Ch2:Display`**
`Ch3:Display` **`Ch4:Display`**

Type: On/off

Action: Turns display of the corresponding channel's signal on or off

Acquire a Channel

Header: **`Ch1:AcqOnlyEn`** **`Ch2:AcqOnlyEn`**
`Ch3:AcqOnlyEn` **`Ch4:AcqOnlyEn`**

Type: On/off

Action: **`On`** - acquisition of the channel is independent of whether it is displayed or not
`Off` - acquisition of the channel occurs only when the channel display is **`On`**

Scale a Channel

Header: **`Ch1:Scale`** **`Ch2:Scale`**
`Ch3:Scale` **`Ch4:Scale`**

Type: Float

Argument: 0.01 to 0.25, or other when attenuator is used

Action: Sets the specified display scale in V/div

Offset a Channel

Header: **`Ch1:Offset`** **`Ch2:Offset`**
`Ch3:Offset` **`Ch4:Offset`**

Type: Float

Argument: –1 to +1, or other when attenuator is used

Action: Sets the specified compensation voltage of the channel in V

Position a Channel

Header: **Ch1:Position** **Ch2:Position**
Ch3:Position **Ch4:Position**

Type: Float

Argument: −5 to +5

Action: Sets the specified vertical position of the channel on the screen, in divisions.

Bandwidth of Channel

Header: **Ch1:Band** **Ch2:Band**
Ch3:Band **Ch4:Band**

Type: Selector

Arguments: **Full, Middle, Narrow**

Action: Sets the bandwidth of the channel

Deskew of Channel

Header: **Ch1:Deskew** **Ch2:Deskew**
Ch3:Deskew **Ch4:Deskew**

Type: Float

Argument: 0 to 100e−9

Action: Sets the deskew of the channel in s

Attenuator linear/log

Header: **Ch1:Atten:Unit** **Ch2:Atten:Unit**
Ch3:Atten:Unit **Ch4:Atten:Unit**

Type: Selector

Arguments: **Off, Ratio, DB**

Action: Sets the presence and scale of the attenuator or converter used with the channel

Attenuator ratio

Header: **Ch1:Atten:Ratio** **Ch2:Atten:Ratio**
Ch3:Atten:Ratio **Ch4:Atten:Ratio**

Type: Float

Argument: 0.0001 to 1000000

Action: Sets the attenuation ratio. This setting is active only when the attenuator unit is ratio.

Attenuator dB

Header: **Ch1:Atten:DB** **Ch2:Atten:DB**
Ch3:Atten:DB **Ch4:Atten:DB**

Type: Float

Argument: -80 to +120

Action: Sets the attenuation in dB. This setting is only active when the attenuator unit is *decibels*.

Attenuator unit

Header: **Ch1:Atten:Dimens** **Ch2:Atten:Dimens**
Ch3:Atten:Dimens **Ch4:Atten:Dimens**

Type: Selector

Arguments: **Volt, Watt, Ampere, Unknown**

Action: Sets the units of the converter used with the channel

4.5 Timebase commands

Sampling Mode

Header: **Instr:TimeBase:SampleModeSet**

Type: Selector

Arguments: **RealTime, RandomET, Roll, Auto**

Action: Sets the instrument's sampling mode

Primary Priority Mode

Header: **TB:Priority:Primary**

Type: Selector-type command

Arguments: **RecLength, SmplRate, HorScale**

Action: Set the primary priority of timebase

Secondary Priority Mode

Header: **TB:Priority:Secondary**

Type: Selector-type command

Arguments: **RecLength, SmplRate, HorScale**

Action: Set the secondary priority of timebase

Timebase scale, sec/div
Header: `Instr:TimeBase:ScaleT`

Type: Float

Argument: PicoScope 9404-05 and PicoScope 9402-05: 50e–12 to 1000;
PicoScope 9404-16 and PicoScope 9402-16: 20e–12 to 1000;

Action: Sets the scale of the timebase

Timebase Sample Rate
Header: `Instr:TimeBase:SmplRate`

Type: Float-type command

Argument: PicoScope 9404-05 and PicoScope 9402-05: 125e-3 to 1e12;
PicoScope 9404-16 and PicoScope 9402-16: 125e-3 to 0.4e12

Action: Sets sample rate in samples per second

Record Length
Header: `Instr:TimeBase:RecLen`

Type: Integer-type command

Argument: 50 to 250000

Action: Sets number of points of signals

Timebase Delay
Header: `TB:Delay`

Type: Float

Argument: 5E-8 to 4.28

Action: Sets the delay of the timebase, in seconds

Trigger Position
Header: `TB:TrigPos`

Type: Float

Argument: 0 to 100

Action: Sets the trigger position, %

4.6 Acquisition commands

4.6.1 Acquisition Mode

Acquisition Mode

Header: **`Acq:Mode`**

Type: Selector-type command

Arguments: **`Sample, Average, EnvMinMax, EnvMin, EnvMax, PeakDetect, HighRes, Segmented`**

Action: Sets the acquisitions mode

Note: Arguments **`PeakDetect, HighRes, Segmented`** are possible in **`RealTime`** sampling mode.

4.6.2 Common Acquisition Commands

General remark on acquisition commands

Some acquisition commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

# of Averaging

Header: **`Acq:NAvg`**

Type: Integer

Argument: 1, 2, 4, 8, 16, ... 4096

Action: Sets the averaging coefficient

# of Envelopes

Header: **`Acq:NEnv`**

Type: Integer

Argument: 2, 4, 8, 16, ... , 4096, 8192

Action: Sets the number of signals for envelope mode. Argument **`8192`** is used for unlimited number of signals

# of High Resolution Bits
Header: **TB:HiResBits**

Type: Float-type command

Argument: 12.5 to 16 with step = 0.5

Action: Sets the effective number of bits in **HighRes** Acquisition Mode

Channels for High Resolution
Header: **Acq:HiResChs**

Type: Group-on/off-type command

Arguments: **Ch1, Ch2, Ch3, Ch4**

Action: Selects channels to increase the effective number of bits

4.6.3 Segmented Acquisition

General remark on acquisition commands
Some acquisition commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:
- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Max Number of Segments query
Header: **Acq:Segment:MaxNSeg?**

Type: Integer-type command

Argument: none

Action: Gets max number of segments for current channel's count and record length

Number of Segments
Header: **Acq:Segment:NSegments**

Type: Integer-type command

Argument: 1 to 1024

Action: Sets number of segments. Max value may be less than 1024 for current channel's count and record length

Segments Display Channel

Header: **`Acq:Segment:Source`**

Type: Selector-type command

Argumenst: **`Ch1, Ch2, Ch3, Ch4`**

Action: Sets the channels for display segments

Segments View Mode

Header: **`Acq:Segment:ViewMode`**

Type: Selector-type command

Arguments: **`Off, Overlay, OverSel`**

Action: Sets the mode of display segments

Selected Segment

Header: **`Acq:Segment:SelectedSeg`**

Type: Integer-type command

Argument: 1 to 1024

Action: Setects the segment for highlighting. Max value may be less than 1024 for current channel's count and record length

Range of segments for overlays

Header: **`Acq:Segment:FirstSegm`**
`Acq:Segment:LastSegm`

Type: Integer-type command

Argument: 1 to 1024

Action: Selects the range of segments for overlays. Max value may be less than 1024 for current channel's count and record length

Segments time table

Header: **`Acq:Segment:TimeTable`**

Type: On/off-type command

Action: Shows or Hides the table of segment's times

4.6.4 Termination of the Acquisition

Termination of Acquisition

Header: **`Acq:RunUntil`**

Type: Selector

Arguments: **`StopBtn, NAcq`**

Action: Sets the condition for terminating acquisition when the Stop button is pressed or after the specified number of waveforms is reached.

4.6.5 Number of Waveforms

Number of Waveforms

Header: **`Acq:NAcq`**

Type: Integer

Argument: 1 to 65535

Action: Sets the number of signals for the terminating acquisition

4.6.6 Action when Number of Waveforms reached

Action when Number of Waveforms reached

Header: **`Acq:Action`**

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

4.6.7 File Name

File Name

Header: **`Acq:FileName`**

Type: Data

Argument: Text string contains the file path

Action: Defines the full path and base file name for storing the acquired signals onto the Disk. The name of each saved file consists of a base name, followed by an underscore (_) and five-digit auto-incremented numbers.

For example, after the command:

Acq:FileName C:\Temp\Test1\basename

Files **basename_00001.wfm, basename_00002.wfm, basename_00003.wfm** and so on will be written to the **C:\Temp\Test1** folder.

Note: The specified folder must exist

4.6.8 Stored Files Format

Stored Files Format
Header: **Acq:FileFormat**

Type: Selector

Argumenst: **Binary, Verbose, YOnly**

Action: Sets the format of the file

4.7 Trigger commands

4.7.1 Trigger

General remark on trigger commands
Some trigger commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following.

- For the PicoScope 9404-05 and the PicoScope 9404-16: mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05: "Trigger" input corresponds to **Ch4** mnemonics;
- For the PicoScope 9402-16: "Direct Trigger" input corresponds to **Ch4** mnemonics;
- For the PicoScope 9402-05 and the PicoScope 9402-16: mnemonic **Ch3** is not used.

Trigger Source
Header: **Trig:Analog:Source**

Type: Selector

Arguments: **CH1, CH2, CH3, CH4**

Action: Sets trigger source

Trigger Style
Header: **Trig:Analog:Style**

Type: Selector-type command

Arguments: **Edge, Divider, ClkRecovery, IntClock, ExtPrescal**

Action: Sets trigger style.

Trigger Level

Header: **Trig:Analog:Ch1:Level** **Trig:Analog:Ch2:Level**
Trig:Analog:Ch3:Level **Trig:Analog:Ch4:Level**

Type: Float-type command

Argument: −1 to +1

Action: Sets the trigger level for the specified channel, in volts

Trigger Slope

Header: **Trig:Analog:Ch1:Slope** **Trig:Analog:Ch2:Slope**
Trig:Analog:Ch3:Slope **Trig:Analog:Ch4:Slope**

Type: Selector-type command

Arguments: **Pos, Neg, BiSlope**

Action: Sets the slope of trigger for specified channel

Trigger Sensitivity

Header: **Trig:Analog:Ch1:Sensitivity**
Trig:Analog:Ch2:Sensitivity
Trig:Analog:Ch3:Sensitivity
Trig:Analog:Ch4:Sensitivity

Type: Selector-type command

Arguments: **High, Low, Var**

Action: Sets trigger sensitivity for specified channel

Trigger Hysteresis

Header: **Trig:Analog:Ch1:Hyst** **Trig:Analog:Ch2:Hyst**
Trig:Analog:Ch3:Hyst **Trig:Analog:Ch4:Hyst**

Type: Float-type command

Argument: 0 to 100

Action: Sets trigger hysteresis for specified channel in the **Var** Trigger Sensitivity

4.7.2 Trigger Period for Internal Clock Sources

Trigger Period for Internal Clock

Header: **Trig:Analog:IntClkPeriod**

Type: Float

Argument: 2e-6 … 0.0655

Action: Sets the period for the internal clock trigger style in seconds

4.7.3 Trigger Mode and Holdoff commands

Trigger Mode
Header: **Trig:Mode**

Type: Selector

Arguments: **Free, Trig**

Action: Sets Freerun or Triggered mode for the trigger

Trigger Sharing
Header: **Trig:Shared**

Type: Selector-type command

Arguments: **Shared, Independ**

Action: Sets sharing mode

Holdoff Mode
Header: **Trig:HoldoffBy**

Type: Selector-type command

Arguments: **Time, Random**

Action: Sets Holdoff mode

Holdoff Time
Header: **Trig:HoldoffTime**

Type: Float-type command

Argument: 0.5e-6…15

Action: Sets the holdoff time, in seconds

Holdoff Time for Random mode
Header: **Trig:HoffRandMin**
Trig:HoffRandMax

Type: Float-type command

Arguments: 0.5e-6…15

Action: Sets the minimum and maximum holdoff times for Random mode, in seconds

4.8 Display commands

General remark on display commands
Some display commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Mnemonic **<src>** in some Display Commands signifies Source
(**<src>** is: **Ch1, Ch2, Ch3, Ch4, 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, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4, XY**

Action: Selects the active trace for **AllLocked** trace mode

Display Persistence

Header: **Displ:<src>:Persistence**

Type: Selector-type command

Arguments: **Simple, VarPersist, InfinPers, VarGrayScal, InfGrayScal, VColorGrad, IColorGrad**

Action: In **PerTrace** mode, sets display persistence for the specified trace
In **AllLocked** mode, sets display persistence for all traces

Display Style

Header: **Displ:<src>:Style**

Type: Selector

Arguments: **Dots, Vectors**

Action: In **PerTrace** mode, sets the display style for specified trace
In **AllLocked** mode, sets the display style for all traces

Persistence Time (for VarPersist Style)

Header: **Displ:<src>:PersistTime**

Type: Float

Argument: 0.1 to 20

Action: In **PerTrace** mode, sets the persistence time for the specified trace, in seconds
In **AllLocked** mode, sets the persistence time for all traces, in seconds

Refresh Time (for VarGrayScal or VColorGrade Styles)

Header: **Displ:<src>:RefreshTime**

Type: Float

Argument: 1 to 200

Action: In **PerTrace** mode, sets the refresh time for specified trace, in seconds
In **AllLocked** mode, sets the refresh time for all traces, in seconds

Reset Display Style

Header: **Displ:ResetAll**

Type: Execution

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

Tandem Display Format

Header: **Displ:TwoColumns**

Type: On/off-type command

Action: Turns on or turn off the two columns display mode.

Display Format

Header: **Displ:Format**

Type: Selector

Arguments: **Auto, YT, 2YT, 4YT, XY, CombYTXY, Comb2YTXY**

Action: Selects the number and kinds of screens

Define Trace Screen (for 4YT Format)

Header: **Displ:Screen4:<trace>,**
when **<trace>** is **<src>** or **Hist**

Type: Selector

Arguments: **1, 2, 3, 4**

Action: Moves the specified trace onto the specified screen in 4YT format

Define Trace Screen (for 2YT, Comb2YTXY Formats)

Header: **Displ:Screen2:<trace>,**
when **<trace>** is **<src>** or **Hist**

Type: Selector

Arguments: **1, 2**

Action: Moves the specified trace onto the specified screen in 2YT or Comb2YTXY formats

Source of X Axis for XY Screen

Header: **Displ:XAxis**

Type: Selector

Arguments: **<src>, exclude XY, DB**

Action: Sets the specified signal as X axis for XY screen

Source of Y Axis for XY Screen

Header: **Displ:YAxis**

Type: Selector

Arguments: **<src>, exclude XY, DB**

Action: Sets the specified signal as Y axis for XY screen

Graticule Type

Header: **Displ:Gratic**

Type: Selector

Arguments: **Grid, Frame, Axis, Off**

Action: Defines the type of graticule for YT and XY screens

Large Dots Display Mode

Header: **Displ:Dot3x3**

Type: On/off-type command

Action: Turns on or turn off the large dots display mode. Used only with short signal length.

Visibility of Main Toolbar

Header: **`Displ:Toolbar`**

Type: On/off-type command

Action: Turns on or turn off the main toolbar

Visibility of Permanent Controls

Header: **`Displ:Permanent`**

Type: On/off-type command

Action: Turns on or turns off the permanent controls

Visibility of Measurements Area

Header: **`Displ:MeasArea`**

Type: On/off-type command

Action: Turns on or turn off the measurements area

Visibility of Side Menu Panels

Header: **`Displ:SideMenu`**

Type: Selector-type command

Arguments: **`Left, Right, Both, None`**

Action: Sets the specified mode of side menus visibility

4.9 Save/Recall commands

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

General remark on save/recall commands
Some save/recall commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Memory Display

Header: `Save:<mz>:Visible`

Type: On/off-group

Items: `M1, M2, M3, M4`

Action: Controls the display of memory zones

Source for storing into Memory

Header: `Save:Memo:Source`

Type: Selector

Arguments: `Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4`

Action: Defines the signal as source for storing into memory zone

Save into Memory

Header: `Save:<mz>:Save`

Type: Execution

Action: Stores the selected source into selected memory

4.9.2 Work with Disk

General remark on save/recall commands
Some save/recall commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

File Type

Header: `Save:Disk:FileType`

Type: Selector

Arguments: `Wfm, DB`

Action: Defines the file type for saving

Source for saving to file

Header: `Save:Disk:Source`

Type: Selector

Arguments: `Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4`

Action: Defines the 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: Defines the file name for saving the specified signal to disk

Note: Specified folder must exist

File Name Mode

Header: `Save:Disk:NameMode`

Type: Selector

Arguments: `Manual, Auto`

Action: Sets the 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: Sets the file format

Save to Disk
Header: **Save:Disk:ExecSave**

Type: Executing

Action: Saves the selected source to previously specified file

Load from Disk
Header: **Save:<mz>:LoadFromDsk**

Type: Executing

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

4.9.3 Work with Setups

General remark on save/recall commands
Some save/recall commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:
- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Recall Factory Setup
Header: **Save:Setup:RecFact**

Type: Execution

Action: Returns the instrument to manufacturer's default setting

Recall Default Setup
Header: **Save:Setup:RecDefault**

Type: Executing

Action: Returns the instrument to its default setting

Recall Power-Off Setup
Header: **`Save:Setup:RecLast`**

Type: Execution

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

Save Setup as Default
Header: **`Save:Setup:SvAsDefault`**

Type: Execution

Action: Stores the present front-panel setup as the default setup

Name of Custom Setup File
Header: **`Save:Setup:FileName`**

Type: Data

Argument: Text string containing file path

Action: Defines the file name for storing Custom Setup

Note: The specified folder must exist

Save Custom Setup
Header: **`Save:Setup:Save`**

Type: Execution

Action: Stores the present front-panel setup as previously specified custom setup

Recall Custom Setup
Header: **`Save:Setup:Recall`**

Type: Execution

Action: Recalls the setup previously saved to file. The name of the setup must first be defined by the command **`Save:Setup:FileName`**.

4.10 Markers commands

General remark on markers commands
Some markers commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Marker Type

Header: `Mark:Type`

Type: Selector

Arguments: `Off, MX, MY, XY`

Action: Sets the marker type

Marker Sources

Header: `Mark:M1:Source` `Mark:M2:Source`

Type: Selector

Arguments: `Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4`

Action: Attaches the specified marker to the specified signal

X position of Marker

Header: `Mark:M1:XPos` `Mark:M2:XPos`

Type: Float

Argument: Real value of X-axis

Action: Sets the X position of the specified marker

Y position of Marker

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

Type: Float

Argument: Real value of Y-axis

Action: Sets the Y position of the 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

4.11.1 Measurements of Time Domain Signals

General remark on measure commands
Some measure commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:
- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

The mnemonic **<src>** in some Measure Commands signifies the Source
(**<src>** is: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**)

Measurement Type
Header: **Meas:Display**

Type: Selector

Arguments: **Off, Param, Statistic**

Action: Sets the measurement type

Measurement Source
Header: **Meas:DisplSrc**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

Action: Sets the source for the measurement

Measurement Mode
Header: **`Meas:Mode`**

Type: Selector

Arguments: **`Permanent, Single`**

Action: Sets the measurement mode

Execute Single Measurement
Header: **`Meas:SingleMeas`**

Type: Execution

Action: Executes a single measurement in **`Single`** mode

4.11.2 Statistic Commands

Statistic Measurement Mode
Header: **`Meas:StatMode`**

Type: Selector

Arguments: **`Permanent, Window, Weight`**

Action: Sets the Statistic Measurement mode

Windows Value
Header: **`Meas:Window`**

Type: Integer

Argument: 8 to 8192

Action: Sets the number of recently acquired waveforms for **`Window`** mode of Statistic Measurement

Weight Value
Header: **`Meas:Weight`**

Type: Integer

Argument: 8 to 8192

Action: Sets the weight variable for **`Weight`** mode of Statistic Measurement

4.11.3 Define Parameter Commands

Viewing of Define Parameters
Header: **`Meas:View`**

Type: On/off-type command

Action: Sets the visibility of *define parameters* markers for selected sources

Top/Base Definition Method
Header: **`Meas:<src>:Method`**

Type: Selector

Arguments: **`Hist, MinMax, Marker`**

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

Top Value for Marker Method
Header: **`Meas:<src>:Top`**

Type: Integer

Argument: 257 to 1023

Action: Sets the 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 767

Action: Sets the 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 the 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 the units of thresholds for the specified signals. 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

Argument: -80 to +200

Action: Sets the threshold percentage for the specified signals. Used for Percent threshold units only. Argument 0 (%) corresponds 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 the margins definition mode

Slope of Left or Right Margins

Headers: **Meas:<src>:LeftSlope**
Meas:<src>:RightSlope

Type: Integer

Argument: 0 to 127

Action: Sets the margin for the specified signals on the specified slope. Used for **slope** margins definition mode only. Argument 0 = the first rise, value 1 = 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. Used for slope margins definition mode only.

Position of Left or Right Margin

Headers: **Meas:<src>:LeftMarker**
Meas:<src>:RightMarker

Type: Float

Argument: Absolute time value

Action: Sets the position of margin for the specified signals. Used for marker margins definition mode only.

4.11.4 List of X Measurements

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: Defines the set of X-axis measurements for the specified signals

4.11.5 List of Y Measurements

List of Y Measurements

Header: **Meas:<src>:YParam**

Type: On/off-group

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

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

4.11.6 Second Source for Inter-Signal Measurements

General remark on measure commands

Some measure commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Second Source for Inter-Signal Measurements

Header: **Meas:Source2**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

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

4.11.7 List of Inter-Signal Measurements

List of Inter-Signal Delay Measurements

Header: **Meas:<src>:DualParDelay**

Type: On/off-group

Items: **Del1R1R, Del1R1F, Del1F1R, Del1F1F, Del1RnR, Del1RnF, Del1FnR, Del1FnF**

Action: Defines the set of the inter-signal delay measurements for the specified signal

List of Inter-Signal Phase Measurements

Header: **Meas:<src>:DualParPhase**

Type: On/off-group

Items: **PhaseDeg, PhaseRad, PhasePerc**

Action: Defines the set of the inter-signal phase measurements for the specified signal

List of Inter-Signal Gain Measurements

Header: **Meas:<src>:DualPar**

Type: On/off-group

Items: **Gain, DBGain**

Action: Defines the set of the inter-signal gain measurements for the specified signal

4.11.8 Measurements of Spectrum Signals

Mnemonic **<fft_src>** in some Measurement Commands signifies Source (**<fft_src>** is **F1, F2, F3, F4, M1, M2, M3, M4**)

4.11.8.1 Spectrum Parameter Commands

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. 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. 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. 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

4.11.8.2 List of Spectrum Frequency Measurements

List of Spectrum Frequency Measurements
Header: **`Meas:<src>:XFFTPar`**

Type: On/off-group

Items: **`Freq, DFreq`**

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

4.11.8.3 List of Spectrum Magnitude Measurements

List of Spectrum Magnitude Measurements
Header: **`Meas:<src>:YFFTPar`**

Type: On/off-group

Items: **`Magn, DMagn, TDH`**

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

4.11.9 Delete all Measurements for all Sources

Delete all Measurements for all Sources
Header: **`Meas:ClearAll`**

Type: Execution

Action: Clears the list of all measurements for all signals

4.11.10 Getting Measurement Results

Get List of Measured Parameters

Header: **`Meas:Res:List?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns 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: Returns 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

Argument: None

Forms: Command with query only

Action: Returns the specified statistic parameter of the measured parameter

4.12 Limit Test commands

4.12.1 Limit Test On/Off

Limit Test On/Off
Header: **`Limit:TestOn`**

Type: On/off

Action: Enables/disables the Limit Test. Must be set **`On`** after full definition of all other Limit Test parameters.

4.12.2 Limit Test Termination Commands

Limit Test Termination Condition
Header: **`Limit:RunUntil`**

Type: Selector

Arguments: **`StopBtn, Failur, Wfm`**

Action: Sets the condition of Limit Test Termination

Number of Failures
Header: **`Limit:Failures`**

Type: Integer

Argument: 1 to 10000

Action: Sets number of failures for the **`Failur`** Condition of the Limit

Number of Waveforms
Header: **`Limit:NWfms`**

Type: Integer

Argument: 1 to 1000000

Action: Sets the number of waveforms for the **`Wfm`** Condition of the Limit

4.12.3 Limit Test Action Commands

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: Sets the file format

File Name

Header: **Limit:FileName**

Type: Data

Argument: Text string

Forms: Command, query, command with query

Action: Defines the file name for saving the specified signals to disk

4.12.4 Parameter Definition Commands

Parameter Activity

Headers: **Limit1:Activ** **Limit2:Activ**
Limit3:Activ **Limit4:Activ**

Type: On/off

Action: Enables/disables the Limit Test for relevant parameter

Parameter Limit Mode

Headers: **Limit1:Mode** **Limit2:Mode**
Limit3:Mode **Limit4:Mode**

Type: Selector

Arguments: **Center, Limit**

Action: **Sets the 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

Argument: Absolute value of limit

Action: Sets the limit's value. 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: Sets the mode of the center definition for the relevant parameter. Used only for the **Center** mode of the parameter limit.

Center Value

Headers: **Limit1:CenterVal** **Limit2:CenterVal**
Limit3:CenterVal **Limit4:CenterVal**

Type: Float

Argument: Absolute value of center

Action: Sets the absolute center value. 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: Sets the mode of delta definition for relevant parameter. 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

Argument: 0.1 to 100 standard deviations of the parameter

Action: Sets the delta value. Used for **StdDev** mode of parameter delta only.

Parameter Delta Value for User Defined Mode

Headers: **Limit1:UserDef** **Limit2:UserDef**
Limit3:UserDef **Limit4:UserDef**

Type: Float

Argument: Absolute value of delta

Action: Sets the delta value. 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

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

Action: Sets the delta value. 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: Sets 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: Sets the limit status when measurement is undefined

4.13 Mathematics commands

4.13.1 Enable Mathematical Function

Enable Mathematical Function

Headers: **F1:On** **F2:On**
F3:On **F4:On**

Type: On/off

Action: Enables/disables the calculation and display of the relevant functions

4.13.2 Display Mathematical Function

DisplayMathematical Function

Headers: **F1:Display** **F2:Display**
F3:Display **F4:Display**

Type: On/off

Action: Enables/disables the visibility of the relevant functions

4.13.3 Function Category

Function Category

Headers: **F1:Category** **F2:Category**
F3:Category **F4:Category**

Type: Selector

Arguments: **Arithm, Algebra, Trigonom, FFT, BitOp, Misc, Formula**

Action: Sets the category of the specified function

4.13.4 Function Operators

Arithmetic Function Operator

Headers: **F1:ArithmOp** **F2:ArithmOp**
F3:ArithmOp **F4:ArithmOp**

Type: Selector

Arguments: **Add, Subtract, Multiply, Divide, Ceil, Floor, Fix, Round, Absolute, Invert, Common, ReScale**

Action: Sets the operator of the specified function. Used for **Arithm** category only.

Algebraic Function Operator

Headers: **F1:AlgebraOp** **F2:AlgebraOp**
F3:AlgebraOp **F4:AlgebraOp**

Type: Selector

Arguments: **ExpE, LogE, Exp10, Log10, ExpA, LogA, Differentiate, Integrate, Square, SqRoot, Cube, PowerA, Inverse, SqRtOfSum**

Action: Sets the operator of the specified function. Used for **Algebra** category only.

Trigonometric Function Operator

Headers: **F1:TrigonOp** **F2:TrigonOp**
F3:TrigonOp **F4:TrigonOp**

Type: Selector

Arguments: **Sine, ASine, Cosine, ACosine, Tangent, ATangent, Cotangent, ACotangent, HSine, HCosine, HTangent, HCotangent**

Action: Sets the operator of the specified function. Used for **Trigonom** category only.

FFT Function Operator

Headers: **F1:FFTOp** **F2:FFTOp**
F3:FFTOp **F4:FFTOp**

Type: Selector

Arguments: **FFT, IFFT, FFTMagn, FFTPhase, FFTReal, FFTIm**

Action: Sets the operator of the specified function. Used for **FFT** category only.

Bits Function Operator

Headers: **F1:BitOp** **F2:BitOp**
F3:BitOp **F4:BitOp**

Type: Selector

Arguments: **And, NAnd, Or, NOr, XOr, NXOr, Not**

Action: Sets the operator of the specified function. Used for **BitOp** category only.

Miscellaneous Function Operator

Headers: **F1:MiscOp** **F2:MiscOp**
F3:MiscOp **F4:MiscOp**

Type: Selector

Arguments: **LinInterp, SinXInterp, Trend, Smooth**

Action: Sets the operator of the specified function. Used for **Misc** category only.

4.13.5 Function Operands

General remark on mathematics commands
Some mathematics commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Operand 1

Headers: **F1:Source1** **F2:Source1**
F3:Source1 **F4:Source1**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

Action: Sets the first operand of the specified function

Operand 2

Headers: **F1:Source2** **F2:Source2**
F3:Source2 **F4:Source2**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4, Constant**

Action: Sets the second operand of the specified function. Used for dual- or quad-operand function.

Operands 3/4

Headers: **F1:Source3** **F1:Source4**
F2:Source3 **F2:Source4**
F3:Source3 **F3:Source4**
F4:Source3 **F4:Source4**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

Action: Sets the third and fourth operands for the specified function. Used for bits function.

Constant Value

Headers: **F1:Const** **F2:Const**
F3:Const **F4:Const**

Type: Float

Arguments: Absolute value of constant

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

4.13.6 Additional Parameters for Arithmetic Functions

Rounding Step

Headers: **F1:RoundTo** **F2:RoundTo**
F3:RoundTo **F4:RoundTo**

Type: Float

Arguments: Value of rounding step

Action: Sets the step for rounding function. Used for **Ceil, Floor, Fix, Round** arithmetic functions.

Rescale Parameters

Headers: **F1:ResMult** **F1:ResOffset**
F2:ResMult **F2:ResOffset**
F3:ResMult **F3:ResOffset**
F4:ResMult **F4:ResOffset**

Type: Float

Arguments: Value of Mult and Offset parameters

Action: Sets the Mult and Offset parameters. Used for **ReScale** arithmetic function.

4.13.7 Additional Parameters for Algebraic Functions

Logarithmic Base

Headers: **F1:LogBase** **F2:LogBase**
F3:LogBase **F4:LogBase**

Type: Float

Arguments: 1.01 to 100

Action: Sets the logarithmic base for **LogA** algebraic function

Number Exponent

Headers: **F1:PowerExp** **F2:PowerExp**
F3:PowerExp **F4:PowerExp**

Type: Float

Argument: −100 to +100

Action: Sets the Number Exponent for **ExpA** algebraic function

4.13.8 Additional Parameters for Trigonometric Functions

Volt-to-Radian Coefficient

Headers: **F1:YScaleRad** **F2:YScaleRad**
F3:YScaleRad **F4:YScaleRad**

Type: Float

Arguments: −100 to +100

Action: Sets the volt-to-radian coefficient for **att** trigonometric functions

4.13.9 Additional Parameters for FFT Functions

Window

Headers: **F1:Window** **F2:Window**
F3:Window **F4:Window**

Type: Selector

Arguments: **Rectang, Hamming, Hanning, FlatTop, BlackHarr, KaiserBess**

Action: Sets the window for the specified function

Suppression

Headers: **F1:Suppress** **F2:Suppress**
F3:Suppress **F4:Suppress**

Type: Group-on/off

Arguments: **DC, PHASE**

Action: **DC** - on/off the suppression of the spectrum DC component;
PHASE - on/off the suppression of the spectrum phase noise.

F1:SupprLevel: Float (–120 to –10, dB)

Phase Suppression Level

Headers: **F1:SupprLevel** **F2:SupprLevel**
F3:SupprLevel **F4:SupprLevel**

Type: Float

Arguments: –120 to –10, dB

Action: Sets the phase suppression level with respect to a maximum magnitude

4.13.10 Additional Parameters for Bit Functions

Source Thresholds

Headers: **F1:Thresh1** **F2:Thresh1**
F3:Thresh1 **F4:Thresh1**

F1:Thresh2 **F2:Thresh2**
F3:Thresh2 **F4:Thresh2**

F1:Thresh3 **F2:Thresh3**
F3:Thresh3 **F4:Thresh3**

F1:Thresh4 **F2:Thresh4**
F3:Thresh4 **F4:Thresh4**

Type: Float

Arguments: Value of thresholds

Action: Sets the threshold levels for each source of the bit functions

Source Inversion

Headers: **F1:SorceInvert** **F2:SorceInvert**
F3:SorceInvert **F4:SorceInvert**

Type: Group-on/off

Arguments: **SRC1, SRC2, SRC3, SRC4**

Action: Enables/disables the inversion of each source

4.13.11 Additional Parameters for Miscellaneous Functions

Smoothing Parameter

Headers: **F1:SmoothLen** **F2:SmoothLen**
F3:SmoothLen **F4:SmoothLen**

Type: Integer

Argument: 0 to 24

Action: Sets the length of the smoothing interval in points for the specified function. Used for **Smooth** operator only. Length is defined as 3 + <Argument> * 2.

Signal Length

Headers: **F1:SignalLen** **F2:SignalLen**
F3:SignalLen **F4:SignalLen**

Type: Integer-type command

Argument: 4000 to 8192

Action: Sets the length of the interpolation function signal. Used for **LinInterp** and **SinXInterp** functions.

Trend Measurement

Headers: **F1:TrendMeas** **F2:TrendMeas**
F3:TrendMeas **F4:TrendMeas**

Type: Selector

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

Action: Sets the kind of trends for the specified function. Used for **Trend** operator only.

4.13.12 Function Scaling

Complex Format

Header: **F1:ComplexScale** **F2:ComplexScale**
F3:ComplexScale **F4:ComplexScale**

Type: Selector

Arguments: **Magnitude, Phase, Real, Imaginary**

Action: Defines the spectrum display mode for **FFT** function

Vertical Scale Type

Header: **F1:VScaleType** **F2:VScaleType**
F3:VScaleType **F4:VScaleType**

Type: Selector

Arguments: **Linear, Logarithm**

Action: Defines the vertical scale type for Magnitude of the **FFT** function

Vertical linear Scale

Header: **F1:VoltScale** **F2:VoltScale**
F3:VoltScale **F4:VoltScale**

Type: Float

Argument: 1e–6 to 1e6

Action: Defines the vertical scale in volts/div for **Linear** vertical scale type

Vertical linear Offset

Header: **F1:VoltOffset** **F2:VoltOffset**
F3:VoltOffset **F4:VoltOffset**

Type: Float

Argument: 1e-6 to 1e6

Action: Defines vertical offset in volts for **Linear** vertical scale type

Vertical linear Position

Header: **F1:VoltPosit** **F2:VoltPosit**
F3:VoltPosit **F4:VoltPosit**

Type: Float

Argument: –10 to +10

Action: Defines the vertical position in div for **Linear** vertical scale type

Vertical logarithmic Scale

Headers: **F1:DBScale** **F2:DBScale**
F3:DBScale **F4:DBScale**

Type: Float

Argument: 1 to 120

Action: Defines the vertical scale in dB/div for **Logarithm** vertical scale type

Vertical logarithmic Offset

Headers: **F1:DBOffset** **F2:DBOffset**
F3:DBOffset **F4:DBOffset**

Type: Float

Argument: -100 to 100

Action: Defines vertical offset in dB/div for **Logarithm** vertical scale type

Vertical logarithmic Position

Headers: **F1:DBPosit** **F2:DBPosit**
F3:DBPosit **F4:DBPosit**

Type: Float

Argument: −10 to +10

Action: Defines the vertical position in div for **Logarithm** vertical scale type

Vertical Phase Scale

Headers: **F1:PhaseScale** **F2:PhaseScale**
F3:PhaseScale **F4:PhaseScale**

Type: Float

Argument: 5.625 to 360

Action: Defines the vertical scale in °/div for **Phase** display mode

Vertical Phase Offset

Headers: **F1:PhaseOffset** **F2:PhaseOffset**
F3:PhaseOffset **F4:PhaseOffset**

Type: Float-type command

Argument: -180 to 180

Action: Defines vertical offset in ° for **Phase** display mode

Vertical Phase Position

Headers: **F1:PhasePosit** **F2:PhasePosit**
F3:PhasePosit **F4:PhasePosit**

Type: Float

Arguments: −10 to +10

Action: Defines the vertical position in div for **Phase** display mode

4.14 Histogram commands

4.14.1 General Histogram Commands

General remark on histogram commands
Some histogram commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Histogram Axis

Header: **Hist:Axis**

Type: Selector

Arguments: **Off, Vert, Horiz**

Action: Sets the axis of the histogram

Histogram Source

Header: **Hist:Source**

Type: Selector

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

Action: Selects the specified signal as source of the histogram

Histogram Visibility

Header: **Hist:Visible**

Type: On/off

Action: Sets the visibility of the histogram. The acquisition of the histogram proceeds independently of this commands.

4.14.2 Histogram Completion Commands

Histogram Finish Condition
Header: **`Hist:RunUntil`**

Type: Selector

Arguments: **`StopSingle, Wfms, Samples`**

Action: Sets the finish condition for acquiring the histogram

Number of Waveforms for Histogram
Header: **`Hist:NWfm`**

Type: Integer-type command

Argument: 1 to 1000000

Action: Sets the number of signals for the termination of histogram acquisition

Number of Samples for Histogram
Header: **`Hist:NSample`**

Type: Integer-type command

Arguments: 1 to 10000000

Action: Sets the number of samples for the termination of histogram acquisition

4.14.3 Histogram Window Commands

Limit Mode for Histogram Window
Header: **`Hist:Limits`**

Type: Selector

Arguments: **`Paired, Independ`**

Action: Sets the mode of the limits of the histogram window

Limit Unit for Histograms Window
Header: **`Hist:Units`**

Type: Selector

Arguments: **`Absolute, Percent`**

Action: Sets the units of the limits of the 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

Arguments: Real value of the X-axis (for Absolute units)
0% to 100% of the X-axis (for Percent units)

Action: Sets 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

Arguments: Real value of the Y-axis (for Absolute units)
0% to 100% of the Y-axis (for Percent units)

Action: Sets the Y positions of the histogram window

Window Visibility

Header: `Hist:Display`

Type: On/off

Action: Sets the visibility of the window

Set Default Window

Header: `Hist:SetDefWind`

Type: Executing-type command

Action: Sets the default window depending on the axis

4.14.4 Histogram Calculation Commands

Calculation Mode
Header: **`Hist:Mode`**

Type: Selector

Arguments: **`Normal, Exponent`**

Action: Sets the mode of histogram calculation

Weight for Exponential Calculation
Header: **`Hist:Weight`**

Type: Integer-type command

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

Action: Sets the weight coefficient for the **`Exponent`** calculation mode

Reset Calculation
Header: **`Hist:RunReset`**

Type: Execution

Action: Restarts histogram calculation

4.14.5 Histogram Scale Commands

Scale Type
Header: **`Hist:ScaleType`**

Type: Selector

Arguments: **`Linear, Logarith`**

Action: Sets the type of histogram scale

Scale Mode
Header: **`Hist:ScaleMode`**

Type: Selector

Arguments: **`Auto, Manual`**

Action: Sets the mode of histogram scale

Linear Scale of Vertical or Horizontal Histogram

Headers: **Hist:VertScale**
Hist:HorScale

Type: Float

Argument: (10 to 100) %/div

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

Logarithmic Scale of Vertical or Horizontal Histogram

Header: **Hist:VertDBScale**
Hist:HorDBScale

Type: Float

Argument: (6 to 60) dB/div

Action: Sets the scale of the histograms. Used for **Manual** mode and **Logarith** type of scale only.

Logarithmic Offset of Vertical or Horizontal Histogram

Header: **Hist:VertDBOffs**
Hist:HorDBOffs

Type: Float

Arguments: (−60 to 0) dB

Action: Sets the offset of the histograms. Used for **Manual** mode and **Logarith** type of scale only.

4.14.6 Histogram Result Commands

Get Histogram Data

Headers: **`Hist:Data?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns a set of text strings with the pair of numbers (comma-separated). First number in the each pair is the histogram axis value, and second number is the histogram value in this point.

Get Histogram Measure

Header: **`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 Mean ± StdDev region, %
- **`Mean2S`** – number of hints in Mean ± 2StdDev region, %
- **`Mean3S`** – number of hints in Mean ± 3StdDev region, %
- **`Min`** – min. value of signal
- **`Max`** – max. value of signal
- **`Max-Max`** – difference between two values of signal, matched two max of histogram

Type: Data

Argument: None

Forms: Query only

Action: Returns a text string with the value of the specified parameter

4.15 Eye Diagram commands

4.15.1 General Eye Commands

General remark on eye diagram commands
Some eye diagram commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Type of Eye Measurements

Header: `Eye:Measure`

Type: Selector

Arguments: `Off, NRZ, RZ`

Action: Sets the type of eye measurements

Sources for Eye Measurements

Header: `Eye:Source`

Type: Selector

Arguments: `Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, DB`

Action: Sets the source for eye measurements

Number of Waveforms in one Measurement

Header: `Eye:WfmsInCycle`

Type: Integer

Argument: 64, 128, 256, 512, 1024

Action: Sets the number of waveforms in one measurement

4.15.2 Eye Measurements Commands

List of X-Axis NRZ Measurements

Header: **Eye:XNRZParam**

Type: Group-on/off

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

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

List of Y-Axis NRZ Measurements

Header: **Eye:YNRZParam**

Type: Group-on/off

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: Defines the set of Y-axis measurements for NRZ signals

List of X-Axis RZ Measurements

Header: **Eye:XRZParam**

Type: Group-on/off

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

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

List of Y-Axis RZ Measurements

Header: **Eye:YRZParam**

Type: Group-on/off

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: Defines the set of Y-axis measurements for RZ signals

Measurements List Clearing
Header: **`Eye:ClearAllMeas`**

Type: Executing

Action: Clears the list of measurement parameters

4.15.3 Define Parameters Commands

Eye Frame Visibility
Header: **`Eye:DisplayWind`**

Type: On/off

Action: Sets the visibility of the eye frame

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: Sets the zone of the period of the NRZ signal for the top/base calculation

Threshold Definition Mode
Header: **`Eye:ThreshMode`**

Type: Selector

Arguments: **`10-90, 20-80, Custom`**

Action: Sets the mode of threshold definition

Upper and Lower Threshold
Headers: **`Eye:UpThresh`**
`Eye:LowThresh`

Type: Float

Argument: 5% to 95% of amplitude

Action: Sets the thresholds for the slopes calculation. Used for **`Custom`** mode.

4.15.4 Eye Calculation Commands

Measurement Statistic
Header: **`Eye:Statistic`**

Type: On/off

Action: Enables/disables measurement statistics

Measurement Statistic Mode
Header: **`Eye:Mode`**

Type: Selector

Arguments: **`Permanent, Window, Weight`**

Action: Sets the mode of statistics calculation. Used when statistics are enabled.

Windows Value
Header: **`Eye:Window`**

Type: Integer

Argument: 8, 16, 32, …, 8192

Action: Sets the window value. Used for **`Window`** mode of statistics.

Weight Value
Header: **`Eye:Weight`**

Type: Integer

Argument: 8, 16, 32, …, 8192

Action: Sets the weight value. Used for **`Weight`** mode of statistics.

4.15.5 Getting Eye Measurement Results

Get List of Measured Parameters

Header: **`Eye:Res:List?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns a 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: Returns 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: Returns the specified statistical parameter of the measured parameter

4.16 Mask Test commands

4.16.1 Common Mask Test Commands

General remark on mask test commands
Some mask test commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Mask Test On
Header: **`Mask:TestOn`**

Type: On/off

Action: Enables/disables the mask test functionality

Signal for Mask Testing
Header: **`Mask:CompareWith`**

Type: Selector

Arguments: **`Ch1, Ch2, CH3, CH4, F1, F2, F3, F4, DB`**

Action: Selects the signal for mask testing

Actuate Mask Testing
Header: **`Mask:Testing`**

Type: On/off

Action: Enables/disables the comparison with current mask

Mask Erasing
Header: **`Mask:EraseMask`**

Type: Execution

Action: Clears the current mask from the display

4.16.2 Mask Creating

Mask Creating Mode

Header: `Mask:CreateAs`

Type: Selector

Arguments: `Std, Auto, Edit`

Action: Sets the mask creation method

4.16.3 Standard Mask Test Commands

Get List of Standards

Header: `Mask:Std:StdsList?`

Type: Data

Argument: None

Forms: Query only

Action: Returns a list of mask standards with ordinal index

Select Standard

Header: `Mask:Std:StdIndex`

Type: Integer

Argument: 0 to (number of standards–1)

Action: Selects the current standard by its ordinal index

Get List of Masks

Header: `Mask:Std:MasksList?`

Type: Data

Argument: None

Forms: Query only

Action: Returns a list of masks with ordinal index from the selected standard

Select Standard Mask
Header: **`Mask:Std:MaskIndex`**

Type: Integer

Argument: 0 to (number of masks in the current standard –1)

Action: Loads the specified mask by its ordinal index

Alignment of Signal with Standard Mask
Header: **`Mask:Std:Align`**

Type: On/off

Action: Enables/disables the alignment of the tested signal with the standard mask parameters

Enable Margins
Header: **`Mask:Std:MarginsOn`**

Type: On/off

Action: Enables/disables the margin control of eye-typed masks

Margins Value
Header: **`Mask:Std:MarginsVal`**

Type: Float

Arguments: –100% to +100%

Action: Sets the margin's value. Used when margins are enabled

Build Immediately
Header: **`Mask:Std:BuildImmediate`**

Type: On/off

Action: Enables/disables creation of the standard mask immediately after any of its parameters change

4.16.4 Automask Commands

General remark on mask test commands
Some mask test commands use mnemonics or the arguments **Ch1**, **Ch2**, **Ch3** and **Ch4** related to the channels.

These mnemonics for various devices mean the following:

- For the PicoScope 9404-05 and the PicoScope 9404-16, mnemonics **Ch1**, **Ch2**, **Ch3**, **Ch4** mean channels CH1, CH2, CH3 and CH4, respectively;
- For the PicoScope 9402-05 and the PicoScope 9402-16, mnemonics **Ch1**, **Ch2** mean channels CH1, CH2, respectively. Mnemonics **Ch3** and **Ch4** are not used for these models.

Automask Source

Header: **`Mask:Auto:Source`**

Type: Selector

Arguments: **`Ch1, Ch2, CH3, CH4, F1, F2, F3, F4, M1, M2, M3, M4`**

Action: Selects the signal as a template for automask building

Margins Units

Header: **`Mask:Auto:Unit`**

Type: Selector-type command

Arguments: **`Division, Current`**

Action: Selects the margins units for automask building

Automask X-Margins

Header: **`Mask:Auto:DeltaX`**

Type: Float

Arguments: (0.02 to 2) div for **`Division`** margins units real X-axis value for **`Current`** margins units

Action: Sets the X-axis margins around the template signal

Automask Y-Margins

Header: **`Mask:Auto:DeltaY`**

Type: Float

Arguments: (0.03125 to 2) div for **`Division`** margins units real Y-axis value for **`Current`** margins units

Action: Sets the Y-axis margins around the template signal

Automask Build
Header: **`Mask:Auto:BuildAMask`**

Type: Execution

Action: Builds automask immediately

4.16.5 Mask Test Termination

Mask Test Finish Condition
Header: **`Mask:RunUntil`**

Type: Selector

Arguments: **`StopBtn, FailedWfms, FailedSmpls, Wfms, Samples`**

Action: Sets the condition of mask test termination

Number of Failed Waveforms
Header: **`Mask:FailWfms`**

Type: Integer

Argument: 1 to 1000000

Action: Sets the number of failed waveforms for the **`FailedWfms`** finish condition

Number of Failed Samples
Header: **`Mask:FailSmpls`**

Type: Integer-type command

Argument: 1 to 1000000

Action: Sets the number of failed samples for the **`FailedSmpls`** finish condition

Number of Waveforms
Header: **`Mask:NWfms`**

Type: Integer-type command

Argument: 1 to 1000000

Action: Sets the number of waveforms for the **`Wfms`** finish condition

Number of Samples

Header: **Mask:NSamples**

Type: Integer-type command

Argument: 1 to 1000000

Action: Sets the number of samples for the **Samples** finish condition

4.16.6 Mask Test Actions

Select Actions

Header: **Mask:Action**

Type: Group-on/off

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: Sets the 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: Defines the name for storing failed signals on Disk. Used when **Save** action is on.

4.16.7 User Mask

User Masks File Name
Header: **Mask:MaskFile**

Type: Data

Argument: Text string

Forms: Command, query, command with query.

Action: Defines the file name for next loading or saving user mask from the disk

Load User Mask
Header: **Mask:LoadUser**

Type: Execution

Action: Loads the previously specified user mask

Save User Mask
Header: **Mask:SaveUser**

Type: Execution

Action: Saves the current mask as user with previously specified file name

4.16.8 Getting Mask Test 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: Returns a text string with the 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: Returns a text string with the values 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: Returns a text string with the values of failed samples on the margin of specified polygon. Used when **`Margins`** is 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: Returns a text string with the total number of failed samples on the margin and on the specified polygon. Used when **`Margins`** is enabled.

4.17 Autocalibration commands

4.17.1 Single-shot Autocalibration

Start of vertical auto-calibration

Header: **`Flash:Vertical:Autocal:Start`**

Type: Executing-type command

Action: Start self-calibration of Sampler 1

Start of horizontal auto-calibration

Header: **`Flash:TB:Calibr:Autocal`**

Type: Execution

Action: Starts self-calibration of timebase

Get the autocalibration status query

Header: **`Flash:Calibr:AutocalResult?`**

Type: Integer

Action: Command is ignored, and query returns an integer:

−1 Autocalibration in progress.
0 Autocalibration finished OK.
5 Autocalibration failed.

There are the following additional results for vertical auto-calibration.

101…115 There are signals on separate inputs, auto-calibration is interrupted. For normal auto calibration, disconnect the signals from the following channels:

101 CH1;

102 CH2;

103 CH1 & CH2;

104 CH3;

105 CH1 & CH3;

106 CH2 & CH3;

107 CH1 & CH2 & CH3;

108 CH4;

109 CH1 & CH4;

110 CH2 & CH4;

111 CH1 & CH2 & CH4;

112 CH3 & CH4;

113 CH1 & CH3 & CH4;

114 CH2 & CH3 & CH4;

115 CH1 & CH2 & CH3 & CH4.

4.17.2 Periodic Autocalibration

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 after the specified interval
`Temperat` - autocalibration begins when deviation of temperature inside the instrument exceeds the specified value

Note :Periodic autocalibration must be turned off when GUI is in **`RemoteOnly`** or **`Invisible`** state. See GUI command.

Autocalibration Period

Header: **`Util:CalPeriod`**

Type: Float

Argument: 0.5 to 16 hours

Action: Sets the autocalibration period in hours

Temperature Deviation

Header: **`Util:TempChange`**

Type: Float

Argument: 0.5 to 10 °C

Action: Sets the temperature deviation for autocalibration

Get the Temperature of the Instrument Query

Header: **`Calibr:Temperature?`**

Type: Float

Argument: None

Forms: Query only

Action: Returns the temperature inside the device in degrees Celsius

4.17.3 Balancing the channels manually

Balancing channels 1 and 2 manually

Header: **`Flash:Sampler:Ch1:FullBW:Balance`**
`Flash:Sampler:Ch1:NarrowBW:Balance`
`Flash:Sampler:Ch2:FullBW:Balance`
`Flash:Sampler:Ch2:NarrowBW:Balance`

Type: Float

Arguments: −0.5 to 0.5

Action: Query or set the balance value in volts for the specified channel

Balancing channels 3 and 4 manually (PicoScope 9341 only)

Header: **`Flash:Smplr2:Ch3:FullBW:Balance`**
`Flash:Smplr2:Ch3:NarrowBW:Balance`
`Flash:Smplr2:Ch4:FullBW:Balance`
`Flash:Smplr2:Ch4:NarrowBW:Balance`

Type: Float

Arguments: −0.5 to 0.5

Action: Query or set the balance value in volts for the specified channel of Sampler 2

4.18 Waveforms commands

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

Waveform Source

Header: **`Wfm:Source`**

Type: Selector

Arguments: **`Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4`**

Action: Sets the signal to be received

Spectrum Format

Header: **`Wfm:Complex`**

Type: Selector

Arguments: **`Mod, Ph, Re, Im`**

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

Get Number of Points in the Waveform
Header: **`Wfm:Preamb:Poin?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns the number of points in the signal

Get X-Axis Step
Header: **`Wfm:Preamb:XInc?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns 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: Returns 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: Returns the X-axis physical units

Get Y-Axis Unit
Header: **`Wfm:Preamb:YU?`**

Type: Data

Argument: None

Forms: Query only

Action: Returns the Y-axis physical units

4.19 Zoom commands

4.19.1 Common commands for zoom

Creation of new zoom zone
Header: **`Zooms:AddZone`**

Type: Execution

Action: Creates first or next Zoom zone.

Note: Maximum number of zoom zones: 4

Deletion of all Zoom zones
Header: **`Zooms:DelAllZones`**

Type: Execution

Action: Deletes all current zoom zones

Main Graticule size
Header: **`Zooms:MainSignalZone`**

Type: Selector

Arguments: **`msz_1_2, msz_1_4, mszOff`**

Action: Sets the size of the main graticule as ½ of display height, ¼ of display height or erases the main graticule.

Display Mode for two Zoom zones
Header: **`Zooms:Display`**

Type: Selector

Arguments: **`Combine, Separate`**

Action: Sets mode of two Zoom zone displays: **`Combine`** - on the single zoom-graticule, and **`Separate`** – on the different zoom graticules.

4.19.2 Commands for defined zoom zone

Parameter **<Zoom_n>** in Zoom Commands signifies Zoom Zones
(**<Zoom_n>** is: **Zoom1, Zoom2, Zoom3, Zoom4**)

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

Deleting zone
Header: **<Zoom_n>:DelZone**

Type: Execution

Action: Deletes specified zoom zone. If the removed zoom zone was not the last, then the zones behind it occupy the vacated position.

Horizontal Zoom Factor
Header: **<Zoom_n>:HorFactor**

Type: Float

Argument: 1 to 2000

Action: Sets the horizontal zoom factor for the specified zoom zone.

Horizontal Zoom Position
Header: **<Zoom_n>:HorPosition**

Type: Float-type command

Argument: 1 to 100

Action: Sets the horizontal position, %

Vertical Zoom Source
Header: **<Zoom_n>:Source**

Type: Selector-type command

Arguments: **Ch1, Ch2, Ch3, Ch4, F1, F2, F3, F4, M1, M2, M3, M4**

Action: Sets the vertical zoom menu signal

Vertical Zoom Factor
Header: **<Zoom_n>:<src>:VertFactor**

Type: Float

Argument: 0.01 to 100

Action: Sets the vertical zoom factor for specified signal in specified zoom zone.

Vertical Zoom Position
Header: **<Zoom_n>:<src>:VertPosition**

Type: Float

Argument: -8 to 8

Action: Sets the vertical position for specified signal in specified zoom zone, divisions

4.20 Calibrator commands

Calibrator Mode
Header: **InOut:Calibr:Wfm**

Type: Selector

Arguments: **Off, DC, Mndr1k, Freq**

Action: Sets mode of calibrator output: Off, DC level, meander 1 kHz, meander with custom frequency.

DC Mode Voltage
Header: **InOut:Calibr:Voltage**

Type: Float

Argument: –1 to 1

Action: Sets the calibrator voltage in DC mode, in volts

Meander Voltage Mode
Header: **InOut:Calibr:Mode**

Type: Selector

Arguments: **AmplOffset, HighLow**

Action: Selects the method of specifying the amplitude parameters of the meander.

Amplitude of Meander
Header: **InOut:Calibr:Amplitude**

Type: Float

Argument: 0.01 to 2

Action: Sets the amplitude of meander, in volts

Offset of Meander

Header: **`InOut:Calibr:Offset`**

Type: Float

Argument: –0.7 to 0.7

Action: Sets the offset of meander, in volts

High Level of Meander

Header: **`InOut:Calibr:HighLvl`**

Type: Float

Argument: -0.99 to 1

Action: Sets the high level of meander, in volts

Low Level of Meander

Header: **`InOut:Calibr:LowLvl`**

Type: Float-type command

Argument: –1 to 0.99

Action: Sets the low level of meander, in volts

Period of Meander

Header: **`InOut:Calibr:Period`**

Type: Float

Argument: 2e-6 to 0.0655

Action: Sets the period of meander, in seconds

Frequency of Meander

Header: **`InOut:Calibr:Frequency`**

Type: Float

Argument: 15.266 to 5e5

Action: Sets the frequency of meander, Hz

Index

A

C

D

E

F

G

H

I

L

M

O

P

Q

R

S

T

W

Z

