



PicoSource PG900 Series

Fast-Edge Pulse Generators

Programmer's Guide



Contents

1 PicoSource PG900 API reference	1
1 PicoSource PG900 COM server	1
2 ExecCommand method	1
3 COMRC object	1
2 Command 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 Concatenation	4
3 Command classification	6
1 Execution-type commands	6
2 On/off-type commands	6
3 Selector-type commands	6
4 Float-type commands	7
4 Full list of commands	8
1 Header commands	8
2 Output 1 & 2 commands (PG911 or PG914)	8
3 Output 3 & 4 commands (PG912 or PG914)	9
4 Trigger commands	10
1 External trigger commands	10
5 Timing commands	11
6 Display commands	11
7 Setup commands	11
Index	13



1 PicoSource PG900 API reference

The PicoSource PG900 is supplied with an API for any third-party application or library to control the pulse generator and collect signals. The API is *COM-based* and is provided by the PicoSource PG900 GUI application.

1.1 PicoSource PG900 COM server

The COM server implementing the API is called *PicoSource5* and is implemented by the PicoSource PG900 GUI application (*PicoSource5.exe*). It is registered in the system during the setup process, and can be explicitly unregistered and registered again by executing *PicoSource5.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 either the command was 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 *PicoSource5.COMRC*.

2 Command syntax

2.1 Command and query structure

2.1.1 Overview

PicoSource PG900 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:

```
Out1:Enable
```

has a query form:

```
Out1:Enable?
```

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.

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

```
Trig:Source?
```

returns the current source of the trigger, while

```
Trig?
```

returns all the trigger 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
Time:Period?	40 μ s	TIME:PERIOD 40 μ s
Trig:Source?	INTERN	TRIG:SOURCE INTERN

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:

```
TRIG:SOURCE EXTERN
```

```
TRIGger:SOURCE EXTERN
```

```
TRIGgerhgdhdgs:SOURCEblablabla EXTERN
```

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

```
TRIG:SOURCE EXTERN
```

```
TIME:PERIOD 25e-6
```

can be concatenated into the single command

```
TRIG:SOURCE EXTERN; :TIME:PERIOD 25e-6
```

- 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

```
OUT1:OUTPUT ON
```

```
OUT1:AMPLITUDE 3.3
```

into a single command

```
OUT1:OUTPUT ON; AMPLITUDE 3.3
```

The longer version also works equally well:

```
OUT1:OUTPUT ON; :OUT1:AMPLITUDE 3.3
```


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

```
TRIG:SOURCE EXTERN;:OUT1:AMPLITUDE?
```

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:

```
TRIG:SOURCE EXTERN;TIME:PERIOD 25e-6
```

(a colon is needed before TIME)

```
OUT1:OUTPUT ON;:AMPLITUDE?
```

(there is an extra colon before AMPLITUDE. Use OUT1:OUTPUT ON; AMPLITUDE? instead.)

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:

```
*Undo
```

```
*Redo
```

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
```

```
Out1:Output 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:

```
Out1:Output? 1
```

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

3.3 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:

```
EXTERN and INTERN
```

and

```
Display:Horizontal
```

has these possible arguments:

1, 2, 3

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:Source? Extern
```

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

3.4 Float-type commands

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

```
Time:Period 0.01
```

sets the period of the pulses to 10 ms. 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

```
Time:Period? 0.01
```

returns 10 ms, when s is a dimension of the scale, and the prefix m is milli.

The commands

```
Time:Period? 0.1m
```

```
Time:Period? 0.0001
```

```
Time:Period? 1e-4
```

```
Time:Period? 100u
```

are equivalent and set the period of the pulses to the value 100 μ s. All of these commands return 100 μ s.

4 Full list of commands

4.1 Header commands

Header: Header
 Type: On/Off
 Action: Enables/disables headers as part of the query response

4.2 Output 1 & 2 commands (PG911 or PG914)

Enable output

Header: Out1:Enable Out2:Enable
 Type: On/Off
 Action: Turns the pulses on the corresponding outputs on or off

Output amplitude limit

Header: Out1:Limit Out2:Limit
 Type: Float
 Arguments: 2.5 to 6
 Action: Sets the limit of amplitude of the specified output, volts

Output amplitude

Header: Out1:Amplitude Out2:Amplitude
 Type: Float
 Arguments: 2.5 to Limit value
 Action: Sets the amplitude of the specified output, volts

Amplitude control mode

Header: Out:OutsChange
 Type: Selector
 Arguments: Separate, Paired
 Action: Sets the output setup mode

Pulse output performance

Header: Out1:TransitTime Out2:TransitTime
 Type: Selector
 Arguments: Smooth, Fast

Action: Sets the pulse performance of the specified output

Inter-output deskew

Header: OutDeskew:Deskew12

Type: Float

Arguments: $-1e-9$ to $1e-9$

Action: Sets the deskew between output 1 and output 2 in s

4.3 Output 3 & 4 commands (PG912 or PG914)

Enable output

Header: Out3:Enable Out4:Enable

Type: On/Off command

Action: Turns the pulses on the corresponding outputs on or off

Tunnel diode current mode

Header: Out3:ModeTD Out4:ModeTD

Type: Selector

Arguments: Auto, Manual

Action: Sets the mode of tunnel diode current control for the specified output

Tunnel diode head calibration

Header: Out3:ModeRecalibrate
Out4:ModeRecalibrate

Type: Execution

Action: Find the optimal tunnel diode current

Tunnel diode current

Header: Out3:TdCurr Out4:TdCurr

Type: Float

Arguments: 0 to 0.06 for positive head; -0.06 to 0 for negative head

Action: Sets the tunnel diode current in A

Output deskew

Header: Out3:Deskew Out4:Deskew

Type: Float

Arguments: $-500e-12$ to $500e-12$

Action: Sets the deskew of the specified output in s

4.4 Trigger commands

Trigger mode

Header: Trig:Mode

Type: Selector

Arguments: Continuous, Manual

Action: Sets the trigger mode

Single pulse generation (manual trigger mode only)

Header: Trig:ExecShot

Type: Execution

Action: Single pulse will be generated immediately

Period source

Header: Trig:Source

Type: Selector

Arguments: Intern, Extern

Action: Sets the source of the pulse period

4.4.1 External trigger commands

Trigger slope

Header: Trig:Slope

Type: Selector

Arguments: Pos, Neg

Action: Sets positive or negative slope for external trigger input

Trigger level

Header: Trig:Level

Type: Float

Argument: -1 to 1

Action: Sets the trigger level in volts for external trigger input

Holdoff time

Header: Trig:HoldOff

Type: Float

Arguments: 1e-6 to 1
Action: Sets the holdoff time in seconds

4.5 Timing commands

Pulse period (continuous trigger mode only)

Header: Time:Period
Type: Float
Argument: 1e-6 to 1
Action: Sets pulse period of the specified output in seconds

Pulse width

Header: Time:Width
Type: Float
Argument: 1e-9 to 4 e-6
Action: Sets pulse width of the specified output in seconds

4.6 Display commands

Horizontal scale

Header: Display:Horizontal
Type: Selector
Arguments: 1, 2, 3
Action: Sets the one of the three time scales of the display

Vertical graticule mode

Header: Display:Vertical
Type: Selector
Arguments: Merge, Split
Action: Sets common or separated graticules for outputs 1, 2 and outputs 3, 4

4.7 Setup commands

Recall factory setup

Header: Setup:RclFactory
Type: Execution
Action: Returns the instrument to manufacturer's default settings



Index

C

Case-insensitivity 4
 Characteristics 6
 Command classification 6
 Command entry rules 4
 Command header 3
 Command messages 2
 Commands 3
 COMRC object 1, 1
 Concatenation 4

D

Display commands
 select vertical graticule mode 11
 set horizontal time scale 11

E

ExecCommand Method 1
 Execution-type commands 6
 External trigger commands
 set holdoff time 10
 set level 10
 set slope 10

F

Float-type commands 7

H

Header commands
 enable/disable headers 8
 Headers 3

L

Line feeds 4

M

Messages 2
 Mnemonics 4

O

On/Off-type commands 6
 Output 1 & 2 commands (PG911 and PG914)
 acquire a channel 8

attenuator dB 8
 attenuator linear/log 8
 attenuator ratio 8
 attenuator unit 8
 bandwidth of channel 8
 channels commands 8
 display a channel 8
 offset a channel 8
 scale a channel 8
 Output 3 & 4 commands (PG912 and PG914)
 acquire a channel 9
 attenuator dB 9
 attenuator linear/log 9
 attenuator ratio 9
 attenuator unit 9
 bandwidth of channel 9
 display a channel 9
 offset a channel 9
 scale a channel 9
 timebase commands 9

Overview 2

P

PicoSource PG900 COM Server 1

Q

Queries 3
 Query commands 2

S

Selector-type commands 6
 Semicolon 4
 Set commands 2
 Setup commands
 recall factory settings 11
 Specifications 6

T

Timing commands
 set pulse period 11
 set pulse width 11
 Trigger commands
 generate single pulse 10
 set pulse period source 10
 set trigger mode 10

W

White space 4

UK headquarters

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

sales@picotech.com
support@picotech.com

www.picotech.com

USA headquarters

Pico Technology
320 N Glenwood Blvd
Tyler
Texas 75702
United States

Tel: +1 800 591 2796
Fax: +1 620 272 0981