Agilent 3000 Series Oscilloscopes

Programmer's Reference

Agilent Technologies

In This Book

This book is your guide to programming the 3000 Series oscilloscopes.

- Chapter 1, “I/O Module Installation and Configuration” on page 7 contains information on the installation and use of the I/O Module.
- Chapter 2, “Introduction to Programming” on page 13 gives you an introduction to programming the oscilloscopes, along with necessary conceptual information. These chapters describe basic program communications, interface, and syntax.
- Chapter 3, “Commands Quick Reference” on page 19 is a brief listing of the 3000 Series oscilloscope commands and syntax.

Each of the remaining chapters describe the set of commands that belong to an individual subsystem, and explains the function of each command.

- Chapter 4, “Common Commands” on page 27.
- Chapter 5, “Root Level Commands” on page 33.
- Chapter 6, “ACQuire Commands” on page 41.
- Chapter 7, “BEEP Commands” on page 47.
- Chapter 8, “CHANnel<n> Commands” on page 51.
- Chapter 9, “COUNter Commands” on page 61.
- Chapter 10, “DISPlay Commands” on page 65.
- Chapter 11, “KEY Commands” on page 73.
- Chapter 12, “MASK Commands” on page 79.
- Chapter 13, “MEASure Commands” on page 89.
- Chapter 14, “SAVerecall Commands” on page 111.
- Chapter 15, “TIMebase Commands” on page 117.
- Chapter 16, “TRIGger Commands” on page 125.
- Chapter 17, “WAVeform Commands” on page 139.

Contents

1 I/O Module Installation and Configuration

I/O Module Installation 8
I/O Module Configuration 10

2 Introduction to Programming

Communicating with the Oscilloscope 14
Instruction Header 14
Truncation Rule 15
White Space (Separator) 15
Braces 16
Ellipsis 16
Square Brackets 16
Program Message Terminator 16
Block Data 17
Remote Command Tips 17

3 Commands Quick Reference

4 Common Commands

*CLS (Clear Status) 29
*IDN? (Identification Number) 30
*OPC? (Operation Complete) 31
*RST (Reset) 32

5 Root Level Commands

AUTO 35
ForceTrig 36
RUN 37
STOP 38
Trig%50 39

6 ACQuire Commands

AVERages 43
MODE 44
SRATe? 45
TYPE 46

7 BEEP Commands

ENABle 49

8 CHANnel<n> Commands

BWLimit 53
COUPling 54
DISPlay 55
INVert 56
OFFSet 57
PROBe 58
SCALe 59

9 COUNter Commands

ENABle 63
VALue? 64

10 DISPlay Commands

CLEar 67
GRID 68
MENUdisplay 69
PERSistence 70
SCReen 71
TYPE 72

11 KEY Commands

Commands for Front Panel Actions 75
LOCK 77

12 MASK Commands

ENABle 81
OPERate 82
OUTPut 83
SOURce 84
STOPonoutput 85
X 86
Y 87

13 MEASure Commands

CLEar 91
FALLtime 92
FREQuency 93
NDUTycycle 94
NWIDth 95
OVERshoot 96
PDUTycycle 97
PERiod 98
PREShoot 99
PWIDth 100
RISetime 101
VAMPlitude 102
VAVerage 103
VBASe 104
VMAX 105
VMIN 106
VPP 107
VRMS 108
VTOP 109

14 SAVerecall Commands

LOAD 113
LOCation 114
SAVE 115
TYPE 116

15 TIMebase Commands

DELayed 119
HOLDoff 120
POSition 121
SCALe 122
SCAN 123

16 TRIGger Commands

[:EDGE]:COUPling 127
[:EDGE]:LEVel 128
[:EDGE]:SLOPe 129
[:EDGE]:SOURe 130
[:EDGE]:SWEep 131
MODE 132
PULSe:MODE 133
PULSe:WIDTh 134
STATus 135
VIDeo:POLarity 136
VIDeo:STANdard 137
VIDeo:SYNC 138

17 WAVeform Commands

DATA? 141
ERASeofroll? 142
MAXPeakdetect? 143
MEMorydata? 144
MINPeakdetect? 145
SCREENDATA? 146
SCREENMAX? 147
SCREENMIN? 148
SOURce 149
STARtofroll? 150
SYSMemsize? 151
TPOSition? 152
WINDowzoom? 153
WINMemsize? 154
WPOSition? 155
XEND? 156
XINCrement? 157
XORigin? 158
XSTart? 159
YINCrement? 160
YORigin? 161

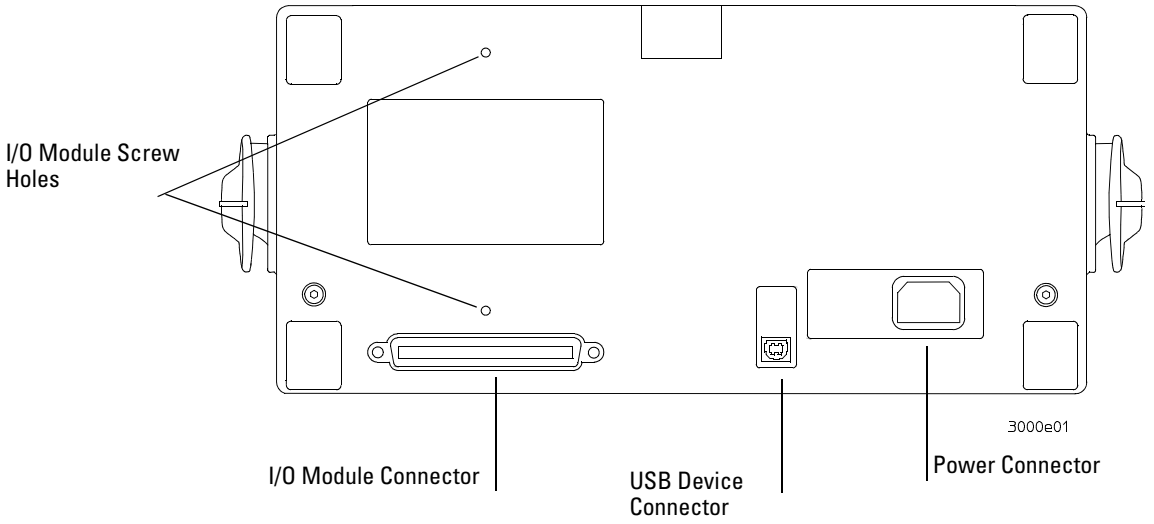
1

I/O Module Installation and Configuration

I/O Module Installation

The I/O module (N2861A) provides an RS-232 serial port and a GPIB port that can be used to remotely program the 3000 Series oscilloscopes. Figure 2 shows the back panel installation location for the I/O module.

Figure 2



Back Panel

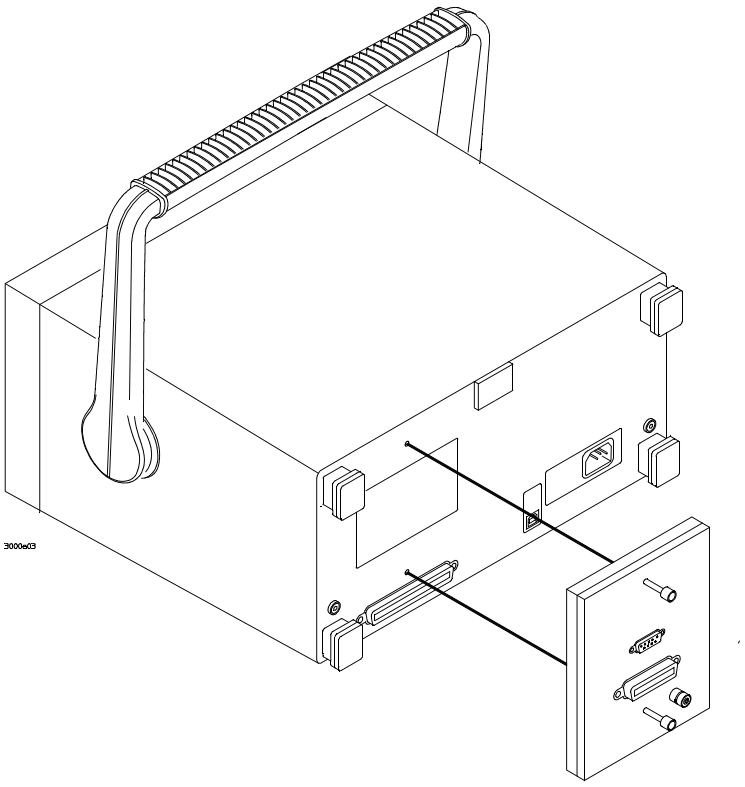
Note

The USB device port is used with Scope Connect software only. This port cannot be used for programming the oscilloscope.

Be sure to turn off the power before installing the I/O module.

Figure 3 shows how to install the I/O module.

Figure 3



Installing the I/O Module

After installation is complete, restart the oscilloscope. The system will automatically detect the I/O module. For example, a message at boot up stating “Communication module installed” means the IO module has been installed, while a message stating “No module installed” means there is no I/O module. In this case, please check that the module is seated correctly.

I/O Module Configuration

After installing the I/O module, the port that you are going to use for remotely programming the oscilloscope must be configured. To configure the I/O module, press the front panel **UTILITY** key. In the UTILITY menu select the **IO Setting** menu key.

RS-232 Configuration

The baud rate can be set to one of the following values:

- 300
- 2400
- 4800
- 9600
- 19200
- 38400

Table 1 RS-232 Data Format

Function	Setting
Data bit	8 bit
Flow control	Not available
Stop bit	1
Odd and even calibration	Not available
Stop character	0D (Hex)

Testing the RS-232 Interface

1 Connect the oscilloscope to a computer using the appropriate RS-232 cable.

2 On the PC, run a terminal-emulator program. Make sure the PC serial port is set as follows:

Table 2 PC RS-232 Setup

Function	Setting
Data bit	8 bit
Flow control	Not available
Stop bit	1
Odd and even calibration	Not available

3 Turn on the oscilloscope.
4 Press front panel **UTILITY** key.
5 Select the I/O Setting menu key.
6 Set the **RS-232 Baud** menu to the same baud rate as that of the PC.
7 Start the terminal-emulator program.
8 Send the *IDN? query.
The oscilloscope responds by returning the identification string of the oscilloscope.
9 Attach a probe between channel 1 and the probe calibration connector.
10 From the terminal-emulator program, send the AUTO command.
The oscilloscope responds by displaying the 3V square wave on the screen.

RS-232 Troubleshooting

If the oscilloscope does not respond, check the following items.

1 Check the I/O module for proper installation.
2 Check the RS-232 cable for proper installation and that the RS-232 cable is connected tightly to the PC and oscilloscope.
3 Check baud rate, data bit, stop bit, flow control, odd and even calibration is correct is correctly set on the PC.
4 Check that the RS-232 baud rate of the oscilloscope is the same as the PC.
5 Verify that the RS-232 cable is correct as shown in Table 3.

Table 3 RS-232 Cable Definition

Pin	Connection
1	NC (No connection)
2	RxD (Receive data)
3	TxD (Transmit data)
4	NC (No connection)
5	GND (Signal ground)
6	NC (No connection)
7	NC (No connection)
8	NC (No connection)
9	NC (No connection)

GPIB Configuration

The GPIB address can be set to any value from 0 to 30.

GPIB Interface Testing

GPIB is the standard for 8 bit parallel communication. The oscilloscope can communicate with a computer, a controller, or a terminal.

The character that is used to terminate a command or query is a new line (0x0A hex) character.

1. Connect the oscilloscope to the computer using a suitable GPIB cable.
2. Run the control terminal software in the computer
3. Turn on the oscilloscope.
4. Press **UTILITY** front panel key.
5. Select the **I/O SETUP** menu key.
6. Set the **GPIB Address** menu to the address you want to use.

   Make sure that this address is exclusive and is not used by any other equipment on the bus.
7. Send the *IDN? query.
8. The oscilloscope responds by returning the identification strings of the oscilloscope.
9. Attach a probe between channel 1 and the probe calibration connector.
10. From the terminal-emulator program, send the AUTO command.

    The oscilloscope responds by displaying the 3V square wave on the screen.

GPIB Troubleshooting

If the oscilloscope does not respond, please check the following items.

1. Check the startup message of the scope. If " No module installed" is displayed, please check that the I/O module is firmly connected.
2. Check the GPIB cable for proper installation and that the GPIB cable is connected tightly to the PC and oscilloscope.
3. Make sure that the GPIB address is correct and exclusive.

2

Introduction to Programming

Introduction to Programming

This chapter introduces the basics for remote programming of an oscilloscope. The programming commands provide the means of remote control.

Basic operations that you can do with a computer and an oscilloscope include:

- Set up the oscilloscope.
- Make measurements.
- Get data (waveforms and measurements) from the oscilloscope.

Communicating with the Oscilloscope

Computers communicate with the oscilloscope by sending and receiving messages over a remote interface, such as a GPIB port or an RS-232 port. Commands for programming normally appear as ASCII character strings embedded inside the output statements of a "host" language available on your computer. The input commands of the host language are used to read responses from the oscilloscope.

Instruction Header

The instruction header is one or more command mnemonics separated by colons (:). They represent the operation to be performed by the oscilloscope. See the "Programming Conventions" chapter for more information.

Queries are formed by adding a question mark (?) to the end of the header. Many instructions can be used as either commands or queries, depending on whether or not you include the question mark. The command and query forms of an instruction usually have different program data. Many queries do not use any program data.

Truncation Rule

The truncation rule is used to produce the short form (abbreviated spelling) for the mnemonics used in the programming commands, queries, and parameter arguments.

Command Truncation Rule

The mnemonic is the first four characters of the keyword, unless the fourth character is a vowel. Then the mnemonic is the first three characters of the keyword. If the length of the keyword is four characters or less, this rule does not apply, and the short form is the same as the long form.

Table 4 shows how the truncation rule is applied to commands.

Table 4 **Mnemonic Truncation**

Long Form	Short Form	How the Rule is Applied
SCALe	SCAL	Short form is the first four characters of the keyword.
TRIGger	TRIG	Short form is the first four characters of the keyword.
AUTO	AUTO	Short form is the same as the long form.
XORigin	XOR	Fourth character is a vowel; short form is the first three characters.

The convention used in this manual to display commands, queries, or parameter arguments is to use upper case letters to indicate the short form.

White Space (Separator)

White space is used to separate the instruction header from the program data. If the instruction does not require any program data parameters, you do not need to include any white space. In this manual, white space is defined as one space. ASCII defines a space to be character 32 in decimal.

Braces

When several items are enclosed by braces, {}, only one of these elements may be selected. Vertical line (|) indicates "or". For example, {ON | OFF} indicates that only ON or OFF may be selected, not both.

Ellipsis

... An ellipsis (trailing dots) indicates that the preceding element may be repeated one or more times.

Square Brackets

Items enclosed in square brackets, [ ], are optional.

Program Message Terminator

The program instructions within a data message are executed after the program message terminator is received. The terminator may be either an NL (New Line) character, an EOI (End-Or-Identify) asserted in the GPIB interface, or a combination of the two. Asserting the EOI sets the EOI control line low on the last byte of the data message. The NL character is an ASCII linefeed (decimal 10).

New Line Terminator Functions Like EOS and EOT

The NL (New Line) terminator has the same function as an EOS (End Of String) and EOT (End Of Text) terminator.

Block Data

Block data is returned as a string representation of hexadecimal values separated by spaces, for example, “0x42 0x43 0x44 ...”. Each hex value in the string represents a data point value. To get the real data value, convert the unsigned hex value to an integer, and use the formula:

(125 – integer data value)YINCrement – YORigin

YINCrement and YORigin are real numbers.

Remote Command Tips

Tip: When writing automated testing routines using the 3000 Series oscilloscope, be sure to use the *OPC? query. The *OPC? query returns a value of '1' when the oscilloscope is finished executing the last command. Waiting for the *OPC? query to return a '1' before issuing the next command ensures that no commands or data are lost.

3

Commands Quick Reference

Commands Quick Reference

The following table provides a quick reference of the commands implemented in the 3000 Series oscilloscopes.

Table 5 Commands Quick Reference

Command	Query	Options and Query Returns
Common Commands		
*CLS	n/a	n/a
n/a	*IDN?	AGILENT TECHNOLOGIES,<model>,<serial number>,XX.XX.XX <model> ::= the model number of the instrument <serial number> ::= the serial number of the instrument <XX.XX.XX> ::= the software revision of the instrument
n/a	*OPC?	1
*RST	n/a	n/a
Root Commands		
:AUTO	n/a	n/a
:ForceTrig	n/a	n/a
:RUN	n/a	n/a
:STOP	n/a	n/a
:Trig%50	n/a	n/a
:ACQuire Commands		
:ACQuire:AVERages <count>	:ACQuire:AVERages?	<count> ::= { 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 }
:ACQuire:MODE <mode>	:ACQuire:MODE?	<mode> ::= { RTIMe | ETIMe}
n/a	:ACQuire:SRATe?	<return_value> ::= NR3 format
:ACQuire:TYPE <type>	:ACQuire:TYPE?	<type> ::= { NORMal | AVERage | PEAK }
:BEEP Commands		
:BEEP:ENABle {{1 | ON} | {0 | OFF}}	:BEEP:ENABle	{ 1 | 0 }

Command	Query	Options and Query Returns
:CHANnel<n> Commands		
:CHANnel<n>:BWLimit { { 1 | ON } | { 0 | OFF } }	:CHANnel<n>:BWLimit?	{ 1 | 0 } <n> ::= 1 - 2
:CHANnel<n>:COUPling { DC | AC | GND }	:CHANnel<n>:COUPling?	{ DC | AC | GND } <n> ::= 1 - 2
:CHANnel<n>:DISPlay { { 1 | ON } | { 0 | OFF } }	:CHANnel<n>DISPlay?	{ 1 | 0 } <n> ::= 1 - 2
:CHANnel<n>INVert { { 1 | ON } | { 0 | OFF } }	:CHANnel<n>INVert?	{ 1 | 0 } <n> ::= 1 - 2
:CHANnel<n>:OFFSet <offset>	:CHANnel<n>:OFFSet?	<offset>::= -8div to +8div <n> ::= 1 - 2
:CHANnel<n>:PROBe <attn>	:CHANnel<n>:PROBE?	<attn>::={ 1 | 10 | 100 | 1000 } <n> ::= 1 - 2
:CHANnel<n>:SCALe <range>	:CHANnel<n>:SCALe?	<range> ::= 2mv to 5v, Probe 1x. 20mv to 50v, Probe 10x. 200mv to 500v, Probe 100x. 2v to 5000v, Probe 1000x. <n> ::= 1 - 2
:COUNter Commands		
:COUNter:ENABle { { 1 | ON } | { 0 | OFF } }	:COUNter:ENABle?	{ 1 | 0 }
n/a	:COUNter:VALue?	<return_value> ::= NR3 format
:DISPlay Commands		
:DISPlay:CLEar	n/a	n/a
:DISPlay:GRID <grid>	:DISPlay:GRID?	<grid> ::= { FULL | HALF | NONE }
:DISPlay:MENUdisplay <time>	:DISPlay:MENUdisplay?	<time> ::= { 1s | 2s | 5s | 10s | 20s | INFinite }
:DISPlay:PERSistence { { 1 | ON } | { 0 | OFF } }	:DISPlay:PERSistence?	{ 1 | 0 }
:DISPlay:SCReen <scr>	:DISPlay:SCReen?	<scr> ::= { NORMal | INVerted }
:DISPlay:TYPE <type>	:DISPlay:TYPE?	<type> ::= { VECTors | DOTS }

Command	Query	Options and Query Returns
:KEY Commands		
:KEY:ACQUIRE	n/a	n/a
:KEY:AUTO_SCALE	n/a	n/a
:KEY:CH1	n/a	n/a
:KEY:CH1_POS_DEC	n/a	n/a
:KEY:CH1_POS_INC	n/a	n/a
:KEY:CH1_SCALE_DEC	n/a	n/a
:KEY:CH1_SCALE_INC	n/a	n/a
:KEY:CH2	n/a	n/a
:KEY:CH2_POS_DEC	n/a	n/a
:KEY:CH2_POS_INC	n/a	n/a
:KEY:CH2_SCALE_DEC	n/a	n/a
:KEY:CH2_SCALE_INC	n/a	n/a
:KEY:CURSOR	n/a	n/a
:KEY:DISPLAY	n/a	n/a
:KEY:F1	n/a	n/a
:KEY:F2	n/a	n/a
:KEY:F3	n/a	n/a
:KEY:F4	n/a	n/a
:KEY:F5	n/a	n/a
:KEY:FORCE	n/a	n/a
:KEY:LOCK { ENABle | DISable }	:KEY:LOCK?	{ENABle | DISable}
:KEY:MAIN_DELAYED	n/a	n/a
:KEY:MATH	n/a	n/a
:KEY:MEASURE	n/a	n/a
:KEY:MNU_ON_OFF	n/a	n/a
:KEY:MODE_COUPLING	n/a	n/a
:KEY:PROMPT_TIME	n/a	n/a
:KEY:REF	n/a	n/a
:KEY:RUN	n/a	n/a
:KEY:SAVE_RECALL	n/a	n/a
:KEY:SINGLE	n/a	n/a
:KEY:TIME_POS_DEC	n/a	n/a
:KEY:TIME_POS_INC	n/a	n/a

Command	Query	Options and Query Returns
:KEY:TIME_SCALE_DEC	n/a	n/a
:KEY:TIME_SCALE_INC	n/a	n/a
:KEY:TRIG_LVL_DEC	n/a	n/a
:KEY:TRIG_LVL_INC	n/a	n/a
:KEY:TRIG%50	n/a	n/a
:KEY:UTILITY	n/a	n/a
:MASK Commands		
:MASK:ENABle{{1 | ON} | {0 | OFF}}	:MASK:ENABle?	{1 | 0}
:MASK:OPERate <opt>	:MASK:OPERate?	<opt> ::= { RUN | STOP }
:MASK:OUTPut <output>	:MASK:OUTPut?	<output> ::= { FAIL | FAIL_SOUND | PASS | PASS_SOUND }
:MASK:SOURce <source>	:MASK:SOURce?	<source> ::= { CHANnel1 | CHANnel2 }
:MASK:STOPonoutput{{1 | ON} | {0 | OFF}}	:MASK:STOPonoutput?	{1 | 0}
:MASK:X <x>	:MASK:X?	<x> ::= 0.4 - 4div
:MASK:Y <y>	:MASK:Y?	<y> ::= 0.4 - 4div
:MEASure Commands		
:MEASure:CLEar	n/a	n/a
:MEASure:FALLtime [<source>]	:MEASure:FALLtime? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:FREQuency[<source>]	:MEASure:FREQuency? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:NDUTycycle [<source>]	:MEASure:NDUTycycle? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:NWIDth [<source>]	:MEASure:NWIDth? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:OVERshoot[<source>]	:MEASure:OVERshoot? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:PDUTycycle [<source>]	:MEASure:PDUTycycle? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:PERiod [<source>]	:MEASure:PERiod? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format

Command	Query	Options and Query Returns
:MEASure:PREShoot [<source>]	:MEASure:PREShoot? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:PWIDth [<source>]	:MEASure:PWIDth? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:RISetime [<source>]	:MEASure:RISetime? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= string
:MEASure:VAMPlitude [<source>]	:MEASure:VAMPlitude? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VAVerage [<source>]	:MEASure:VAVerage? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VBASe [<source>]	:MEASure:VBASe? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VMAX [<source>]	:MEASure:VMAX? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VMIN [<source>]	:MEASure:VMIN? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VPP [<source>]	:MEASure:VPP? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VRMS [<source>]	:MEASure:VRMS? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:MEASure:VTOP [<source>]	:MEASure:VTOP? [<source>]	<source> ::= { CHANnel<n> } n ::= 1 - 2 <return_value> ::= NR3 format
:SAVerecall Commands		
:SAVerecall:LOAD	n/a	n/a
:SAVerecall:LOCation <location>	:SAVerecall:LOCation?	<location> ::= 1 - 10
:SAVerecall:SAVE	n/a	n/a
:SAVerecall:TYPE <type>	:SAVerecall:TYPE?	<type> ::= { WAVeforms | SETups }

Command	Query	Options and Query Returns
:TIMebase Commands		
:TIMebase:DELayed {{1 | ON } | {0 | OFF} }	:TIMebase:DELayed?	{ 1 | 0 }
:TIMebase:HOLDoff <hld_time>	:TIMebase:HOLDoff?	<hld_time> ::= 100ns - 1.5s
:TIMebase:POSition <pos>	:TIMebase:POSition?	<pos> ::= -6div to +6div (50s - 50ms/div) <pos> ::= -14div to +1s (other)
:TIMebase:SCALe <scale_val>	:TIMebase:SCALe?	<scale_val> ::= 1ns-50s/div (DSO3202A) <scale_val> ::= 2ns-50s/div (DSO3152A & DSO3102A) <scale_val> ::= 5ns-50s/div (DSO3062A)
n/a	:TIMebase:SCAN?	{ 1 | 0 }
:TRIGger Commands		
:TRIGger[:EDGE]:COUPling { DC | AC | HF | LF }	:TRIGger[:EDGE]:COUPling?	{ DC | AC | HF | LF }
:TRIGger[:EDGE]:LEVel <level>	:TRIGger[:EDGE]:LEVel?	<level> ::= - 12div to + 12div
:TRIGger[:EDGE]:SLOPe{POSitive | NEGative }	:TRIGger[:EDGE]:SLOPe?	{ POSitive | NEGative }
:TRIGger[:EDGE]:SOURce <src>	:TRIGger[:EDGE]:SOURce?	<src> ::= { CHANnel<n> | EXT | EXT5 | ACLine } n ::= 1 - 2
:TRIGger[:EDGE]:SWEep { AUTO | NORMal }	:TRIGger[:EDGE]:SWEep?	{ AUTO | NORMal }
:TRIGger:MODE <mod>	:TRIGger:MODE?	<mod> ::= { EDGE | PULSe | TV }
:TRIGger:PULSe:MODE <mod>	:TRIGger:PULSe:MODE?	<mod> ::= { +GREaterthan | +LESSthan | + EQUal | -GREaterthan | -LESSthan | -EQUal }
:TRIGger:PULSe:WIDTh <wid>	:TRIGger:PULSe:WIDTh?	<wid> ::= 20ns to 10s
n/a	:TRIGger:STATus?	{ STOP | T'D | WAIT }
:TRIGger:VIDeo:POLarity { POSitive | NEGative }	:TRIGger:VIDeo:POLarity?	{ POSitive | NEGative }
:TRIGger:VIDeo:STANdard { NTSC | PALSecam }	:TRIGger:VIDeo:STANdard?	{ NTSC | PALSecam }
:TRIGger:VIDeo:SYNC <mod>	:TRIGger:VIDeo:SYNC?	<mod> ::= { FIELd<n> | LINe | ALLLines } n ::= 1 - 2

Command	Query	Options and Query Returns
:WAVeform Commands		
n/a	:WAVeform:DATA?	<return_block>
n/a	:WAVeform:ERASeofroll?	<roll_erase_wid> ::= integer in NR1 format
n/a	:WAVeform:MAXPeakdetect?	<return_block>
n/a	:WAVeform:MEMorydata?	<return_block>
n/a	:WAVeform:MINPeakdetect?	<return_block>
n/a	:WAVeform:SCREENDATA?	<return_block>
n/a	:WAVeform:SCREENMAX?	<return_block>
n/a	:WAVeform:SCREENMIN?	<return_block>
:WAVeform:SOURce <source>	:WAVeform:SOURce?	<source> ::= { CHANnel1 | CHANnel2 }
n/a	:WAVeform:STARtofroll?	<roll_start_pos> ::= integer in NR1 format
n/a	:WAVeform:SYSMemsize?	<memory_data_size> ::= integer in NR1 format
n/a	:WAVeform:TPOSition?	<trig_pos> ::= integer in NR1 format
n/a	:WAVeform:WINDowzoom?	<window_zoom> ::= integer in NR1 format
n/a	:WAVeform:WINMemsize?	<window_data_size> ::= integer in NR1 format
n/a	:WAVeform:WPOSition?	<wave_pos> ::= integer in NR1 format
n/a	:WAVeform:XEND?	<screen_wave_endx> ::= integer in NR1 format
n/a	:WAVeform:XINCrement?	<xinc> ::= NR3 format
n/a	:WAVeform:XORigin?	<xorg> ::= NR3 format
n/a	:WAVeform:XSTart?	<sceen_wave_startx> ::= integer in NR1 format
n/a	:WAVeform:YINCrement?	<yinc> ::= NR3 format
n/a	:WAVeform:YORigin?	<yorg> ::= NR3 format

4

Common Commands

Common Commands

Common commands are defined by the IEEE 488.2 standard. They control generic device functions that are common to many different types of instruments. Common commands can be received and processed by the oscilloscope, whether they are sent over the GPIB as separate program messages or within other program messages.

The common commands implemented in the 3000 Series oscilloscopes are:

- "*CLS (Clear Status)" on page 29.
- "*IDN? (Identification Number)" on page 30.
- "*OPC? (Operation Complete)" on page 31.
- "*RST (Reset)" on page 32.

*CLS (Clear Status)

Command

```
*CLS
```

The *CLS command clears all status and error registers.

*IDN? (Identification Number)

Query

```
*IDN?
```

The *IDN? query returns the company name, oscilloscope model number, serial number, and software revision number.

Returned Format

```
AGILENT
TECHNOLOGIES,<model>,<serial_number>,<rev_number><
NL>
```

`<model>` Oscilloscope model number.

`<serial_number>` Specifies the serial number of the oscilloscope. The first four digits and letter are the serial prefix, which is the same for all identical oscilloscopes. The last five digits are the serial suffix, which is assigned sequentially, and is different for each oscilloscope.

`<rev_number>` Specifies the software revision number of the oscilloscope.

*OPC? (Operation Complete)

Query

```
*OPC?
```

The *OPC? query places an ASCII character “1” in the oscilloscope's output queue when all pending selected device operations have finished.

Returned Format

```
1<NL>
```

*RST (Reset)

Command

```
*RST
```

The *RST command places the oscilloscope in a known state. This command loads the Factory setup.

5

Root Level Commands

Root Level Commands

Root level commands control many of the basic operations of the oscilloscope that you can select by pressing the labeled keys on the front panel. These commands are always recognized by the parser if they are prefixed with a colon, regardless of the current tree position. After executing a root level command, the parser is positioned at the root of the command tree.

These root level commands and queries are implemented in the 3000 Series oscilloscopes:

- "AUTO" on page 35.
- "ForceTrig" on page 36.
- "RUN" on page 37.
- "STOP" on page 38.
- "Trig%50" on page 39.

AUTO

Command

```
:AUTO
```

The :AUTO command causes the oscilloscope to evaluate all input waveforms and find the optimum conditions for displaying the waveform. It searches each of the channels for input waveforms and shuts off channels where no waveform is found. It adjusts the vertical gain and offset for each channel that has a waveform, and sets the time base on the lowest numbered input channel that has a waveform.

The trigger is found by first searching external trigger inputs, then searching each channel, starting with channel 1, then channel 2 until a trigger waveform is detected. If waveforms cannot be found on any vertical input, the oscilloscope is returned to its former state.

ForceTrig

Command

```
:ForceTrig
```

The :ForceTrig command starts an start an acquisition even if a valid trigger has not been found. This command has no effect if the acquisition is already stopped.

RUN

Command

```
:RUN
```

The :RUN command starts the oscilloscope running. When the oscilloscope is running, it acquires waveform data according to its current settings. Acquisition runs repetitively until the oscilloscope receives a :STOP command, or until a single acquisition has occurred when the Trigger Sweep is set to Single.

STOP

Command

```
:STOP
```

The :STOP command causes the oscilloscope to stop acquiring data. To restart the acquisition, use the :RUN command.

Trig%50

Command

```
:Trig%50
```

The :Trig%50 command sets the trigger level to the middle of the waveform.

6

ACQuire Commands

ACQuire Commands

The ACQuire subsystem commands set up conditions for acquiring waveform data.

These ACQuire commands and queries are implemented in the 3000 Series oscilloscopes:

- "AVERages" on page 43.
- "MODE" on page 44.
- "SRATe?" on page 45.
- "TYPE" on page 46.

AVERages

Command

```
:ACQuire:AVERages {2 | 4 | 8 | 16 | 32 | 64 | 128 | 256}
```

The :ACQuire:AVERages command sets the number of averages for the waveforms.

Query

```
:ACQuire:AVERages?
```

The :ACQuire:AVERages? query returns the number of averages.

Returned Format

```
{2 | 4 | 8 | 16 | 32 | 64 | 128 | 256}<NL>
```

MODE

Command

```
:ACQuire:MODE {RTIMe | ETIMe}
```

The :ACQuire:MODE command sets the acquisition mode of the oscilloscope.

- The :ACQuire:MODE RTIMe command sets the oscilloscope in real time mode. This mode is useful to inhibit equivalent time sampling at fast sweep speeds.
- The :ACQuire:MODEL ETIMe command sets the oscilloscope in equivalent time mode.

Query

```
:ACQuire:MODE?
```

The :ACQuire:MODE? query returns the acquistion mode of oscilloscope.

Returned Format

```
{RTIM | ETIM}<NL>
```

SRATe?

Command `:ACQuire:SRATe`

Query `:ACQuire:SRATe?`

The :ACQuire:SRATe? query returns the current oscilloscope acquisition sample rate. The sample rate is not directly controllable.

Returned Format `<sample_rate><NL>`

`<sample_rate> ::= sample rate in NR3 format`

TYPE

Command

```
:ACQuire:TYPE {NORMal | AVERage | PEAK}
```

The :ACQuire:TYPE command selects the type of data acquisition that is to take place. The acquisition types are: NORMal, AVERage and PEAK.

- The :ACQuire:TYPE NORMal command sets the oscilloscope in the normal mode.
- The :ACQuire:AVERage command sets the oscilloscope in the averaging mode.
- The :ACQuire:TYPE PEAK command sets the oscilloscope in the peak detect mode.

Query

```
:ACQuire:TYPE?
```

The :ACQuire:TYPE? query returns the current acquisition type.

Returned Format

```
{NORMal | AVERage | PEAK}<NL>
```

7

BEEP Commands

BEEP Commands

The BEEP subsystem commands control all beep functions of the oscilloscope.

These BEEP commands and queries are implemented:

- "ENABle" on page 49.

ENABle

Command `:BEEP:ENABle {{ 1 | ON} | {0 | OFF}}`

The :BEEP:ENABle command enables the audible beep on the oscilloscope.

Query `:BEEP:ENABle?`

The :BEEP:ENABle? query shows whether the audible beep is enabled or disabled.

Returned Format `{1 | 0}<NL>`

8

CHANnel<n> Commands

CHANnel<n> Commands

The CHANnel<n> subsystem commands control all vertical (Y axis) functions of the oscilloscope.

These CHANnel<n> commands and queries are implemented:

- "BWLimit" on page 53.
- "COUPling" on page 54.
- "DISPlay" on page 55.
- "INVert" on page 56.
- "OFFSet" on page 57.
- "PROBe" on page 58.
- "SCALe" on page 59.

BWLimit

Command `:CHANnel<n>:BWLimit {{ 1 | ON} | {0 | OFF}}`

The :CHANnel<n>:BWLimit command controls an internal low-pass filter. When the filter is on, the bandwidth of the specified channel is limited to approximately 25 MHz.

`<n>` 1 or 2

Query `:CHANnel<n>:BWLimit?`

The :CHANnel<n>:BWLimit? query returns the current setting of the low-pass filter.

Returned Format `{1 | 0}<NL>`

COUPling

Command

```
:CHANnel<n>:COUPling {DC | AC | GND}
```

The :CHANnel<n>:DISPlay command turns the display of the specified channel on or off.

The :CHANnel<n>:COUPling command selects the input coupling for the specified channel. The coupling for each channel can be set to AC, DC, or GND.

`<n>` 1 or 2

Query

```
:CHANnel<n>:COUPling?
```

The :CHANnel<n>:COUPling? query returns the current coupling for the specified channel.

Returned Format

```
{DC | AC | GND}<NL>
```

DISPlay

Command `:CHANnel<n>:DISPlay {{ 1 | ON} | {0 | OFF}}`

The :CHANnel<n>:DISPlay command turns the display of the specified channel on or off.

`<n>` 1 or 2

Query `:CHANnel<n>:DISPlay?`

The :CHANnel<n>:DISPlay? query returns the current display condition for the specified channel.

Returned Format `{1 | 0}<NL>`

INVert

Command `:CHANnel<n>:INVert {{ 1 | ON} | {0 | OFF}}`

The :CHANnel<n>:INVert command selects whether or not to invert the input signal for the specified channel. The inversion may be 1 (ON/inverted) or 0 (OFF/not inverted).

`<n>` 1 or 2

Query `:CHANnel<n>:INVert?`

The :CHANnel<n>:INVert? query returns the current state of the channel inversion.

Returned Format `{1 | 0}<NL>`

OFFSet

Command `:CHANnel<n>:OFFSet <offset_value>`

The :CHANnel<n>:OFFSet command sets the voltage that is represented at the center of the display for the selected channel.

`<n>` 1 or 2

`<offset_value>` -8div to +8div

Query `:CHANnel<n>:OFFSet?`

The :CHANnel<n>:OFFSet? query returns the current offset value for the specified channel.

Returned Format `<offset_value><NL>`

PROBe

Command `:CHANnel<n>:PROBe <attn_value>`

The :CHANnel<n>:DISPlay command turns the display of the specified channel on or off.

The :CHANnel<n>:PROBe command specifies the probe attenuation factor for the selected channel. The probe attenuation factor may be 1, 10, 100, or 1000. This command does not change the actual input sensitivity of the oscilloscope. It changes the reference constants for scaling the display factors, for making automatic measurements, and for setting trigger levels.

`<n>` 1 or 2

`<attn_value>` { 1 | 10 | 100 | 1000 }

Query `:CHANnel<n>:PROBe?`

The :CHANnel<n>:PROBe? query returns the current probe attenuation factor for the selected channel.

Returned Format `<attn_value><NL>`

```
<attn_value> ::= { 1 | 10 | 100 | 1000 }
```

SCALe

Command `:CHANnel<n>:SCALe <scale_value>`

The :CHANnel<n>:SCALe command sets the vertical scale, or units per division, of the selected channel. This command is the same as the front-panel channel scale.

`<n>` 1 or 2

`<scale_value>` A number in exponential format for the vertical scale of the channel in units per division. The legal values for the scale range from:

- 2 mV to 5 V when the probe attenuation factor is 1x.
- 20 mV to 50 V when the probe attenuation factor is 10x.
- 200 mV to 500 V when the probe attenuation factor is 100x.
- 2 V to 5000 V when the probe attenuation factor is 1000x.

Query `:CHANnel<n>:SCALe?`

The :CHANnel<n>:SCALe? query returns the current scale setting for the specified channel.

Returned Format `<scale_value><NL>`

`<scale_value> ::= in NR3 format`

9

COUNter Commands

COUNter Commands

The COUNter subsystem commands control all frequency counter functions of the oscilloscope.

These COUNter commands and queries are implemented:

- "ENABle" on page 63.
- "VALue?" on page 64.

ENABle

Command

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

The :COUNter:ENABle command enables the frequency counter.

The frequency counter counts trigger level crossings at the selected trigger slope and displays the results in Hz. The gate time for the measurement is automatically adjusted to be 100 ms or twice the current time window, whichever is longer, up to 1 second. The frequency counter can measure frequencies up to 125 MHz. The minimum frequency supported is 1/(2 X gate time).

The Y cursor shows the the edge threshold level used in the measurement.

Query

```
:COUNter:ENABle?
```

The :COUNter:ENABle? query shows whether the frequency counter is enabled or disabled.

Returned Format

```
{1 | 0}<NL>
```

VALue?

Query

```
:COUNter:VALue?
```

The :COUNter:VALue? query returns the frequency counter value.

Returned Format

```
<counter_value><NL>
<counter_value> ::= in Hz in NR3 format
```

10

DISPlay Commands

DISPlay Commands

The DISPlay subsystem controls the display of data, text, and grids, and the use of color.

These DISPlay commands and queries are implemented in the 3000 Series oscilloscopes:

- "CLEar" on page 67.
- "GRID" on page 68.
- "MENUdisplay" on page 69.
- "PERSistence" on page 70.
- "SCReen" on page 71.
- "TYPE" on page 72.

CLEar

Commands

```
:DISPlay:CLEar
```

The :DISPlay:CLEar command clears the display and resets all associated measurements. If the oscilloscope is stopped, all currently displayed data is erased. If the oscilloscope is running, all of the data for active channels and functions is erased; however, new data is displayed on the next acquisition.

GRID

Command

```
:DISPlay:GRID {FULL | HALF | NONE}
```

The :DISPlay:GRID command selects the type of graticule that is displayed.

- In FULL grid mode, the oscilloscope has a 12-by-8 (unit) display grid, a grid line is place on each vertical and horizontal division.
- In HALF grid mode, only the major horizontal and vertical axes with tic marks are shown.
- When it is off (NONE), a frame with tic marks surrounds the grid edges.

Query

```
:DISPlay:GRID?
```

The :DISPlay:GRID? query returns the current grid setting.

Returned Format

```
{FULL | HALF | NONE}<NL>
```

MENUdisplay

Command

```
:DISPlay:MENUdisplay {1s | 2s | 5s | 10s | 20s |
INFinite}
```

The :DISPlay:MENUdisplay command sets the amount of time that a menu display once activated.

Query

```
:DISPlay:MENUdisplay?
```

The :DISPlay:MENUdisplay? query returns the amount of time that the on screen menu appears when activated.

Returned Format

```
{1s | 2s | 5s | 10s | 20s | INFinite}<NL>
```

PERSistence

Command

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

The :DISPlay:PERSistence command sets the display persistence of waveforms off or on.

- When persistence is OFF, waveforms are erased from the screen at the end of each trigger cycle.
- When persistence is ON, waveforms are not erased with each trigger cycle but accumulates over time.

Query

```
:DISPlay:PERSistence?
```

The :DISPlay:PERSistence? query returns the state of the persistence control.

Returned Format

```
{1 | 0}<NL>
```

SCReen

```
:DISPlay:SCReen {NORMal | INVerted}
```

The :DISPlay:SCReen command sets the color scheme of the display. When set to inverted, display colors are changed to their inverse colors.

Query

```
:DISPlay:SCReen?
```

The :DISPlay:SCReen? query returns the state of the screen control.

Returned Format

```
{NORMal | INVerted}<NL>
```

TYPE

Command

```
:DISPlay:TYPE {DOTS | VECTors}
```

The :DISPlay:TYPE command sets the way that waveforms are drawn. When set to VECTors, waveforms are drawn with lines connecting adjacent sample points. When set to DOTS, only the waveform sample points are drawn.

Query

```
:DISPlay:TYPE?
```

The :DISPlay:TYPE? query returns the state of the type control.

Returned Format

```
{DOTS | VECTors}<NL>
```

11

KEY Commands

KEY Commands

KEY commands control many of the basic operations of the oscilloscope that you can select by pressing the front panel keys.

These KEY commands and queries are implemented in the 3000 Series oscilloscopes:

- "Commands for Front Panel Actions" on page 75.
- "LOCK" on page 77.

Commands for Front Panel Actions

Table 6 KEY Commands

Command:	Is the same as this Front Panel action:
:KEY:ACQUIRE	Pressing the **Acquire** key.
:KEY:AUTO_SCALE	Pressing the **Autoscale** key. The :KEY:AUTO_SCALE command causes the oscilloscope to evaluate all input waveforms and find the optimum conditions for displaying the waveforms. It searches each of the channels for input waveforms and shuts off channels where no waveform is found. It adjusts the vertical gain and offset for each channel that has a waveform, and sets the time base on the lowest numbered input channel that has a waveform. The trigger is found by searching channel 1 then channel 2 until a trigger waveform is detected.
:KEY:CH1	Pressing the **CH1** key.
:KEY:CH1_POS_DEC	Turning the channel 1 **Vertical Position** knob counterclockwise.
:KEY:CH1_POS_INC	Turning the channel 1 **Vertical Position** knob clockwise.
:KEY:CH1_SCALE_DEC	Turning the channel 1 **Vertical Scale** knob counterclockwise.
:KEY:CH1_SCALE_INC	Turning the channel 1 **Vertical Scale** knob clockwise.
:KEY:CH2	Pressing the **CH2** key.
:KEY:CH2_POS_DEC	Turning the channel 2 **Vertical Position** knob counterclockwise.
:KEY:CH2_POS_INC	Turning the channel 2 **Vertical Position** knob clockwise.
:KEY:CH2_SCALE_DEC	Turning the channel 2 **Vertical Scale** knob counterclockwise.
:KEY:CH2_SCALE_INC	Turning the channel 2 **Vertical Scale** knob clockwise.
:KEY:CURSOR	Pressing the **Cursors** key.
:KEY:DISPLAY	Pressing the **Display** key.
:KEY:F1	Pressing the **F1** key.
:KEY:F2	Pressing the **F2** key.
:KEY:F3	Pressing the **F3** key.
:KEY:F4	Pressing the **F4** key.
:KEY:F5	Pressing the **F5** key.
:KEY:FORCE	Pressing the **Force** key.
:KEY:MAIN_DELAYED	Pressing the **Main/Delayed** key.

Command:	Is the same as this Front Panel action:
:KEY:MATH	Pressing the **Math** key.
:KEY:MEASURE	Pressing the **Meas** key.
:KEY:MNU_ON_OFF	Pressing the **MENU ON/OFF** key.
:KEY:MODE_COUPLING	Pressing the **Mode/Coupling** button.
:KEY:PROMPT_TIME	Pressing the **Horizontal Scale** knob.
:KEY:REF	Pressing the **Ref** key.
:KEY:RUN	Pressing the **Run/Stop** key.
:KEY:SAVE_RECALL	Pressing the **Save/Recall** key.
:KEY:SINGLE	Pressing the **Single** key.
:KEY:TIME_POS_DEC	Turning the **Horizontal Position** knob counterclockwise.
:KEY:TIME_POS_INC	Turning the **Horizontal Position** knob clockwise.
:KEY:TIME_SCALE_DEC	Turning the **Horizontal Scale** knob counterclockwise.
:KEY:TIME_SCALE_INC	Turning the **Horizontal Scale** knob clockwise.
:KEY:TRIG_LVL_DEC	Turning the **Trigger Level** knob counter-clockwise.
:KEY:TRIG_LVL_INC	Turning the **Trigger Level** knob clockwise.
:KEY:TRIG%50	Pressing the **Trigger 50%** key.
:KEY:UTILITY	Pressing the **Utility** key.

LOCK

Command

```
:KEY:LOCK {ENABle | DISable}
```

The :KEY:LOCK command enables or disables the front panel.

Query

```
:KEY:LOCK?
```

The :KEY:LOCK? query returns the current state of the front panel lock control.

Returned Format

```
{ENABle | DISable}<NL>
```

12

MASK Commands

MASK Commands

The MASK subsystem controls the Mask Test function.

These MASK commands and queries are implemented in the 3000 Series oscilloscopes:

- "ENABle" on page 81.
- "OPERate" on page 82.
- "OUTPut" on page 83.
- "SOURce" on page 84.
- "STOPonoutput" on page 85.
- "X" on page 86.
- "Y" on page 87.

ENABle

Command `:MASK:ENABle {{1 | ON} | {0 | OFF}}`

The :MASK:ENABle command enables or disables the Mask Test function.

Query `:MASK:ENABle?`

The :MASK:ENABle? query returns the state of the mask enable control.

Returned Format `{1 | 0}<NL>`

OPERate

Command

```
:MASK:OPERate {RUN | STOP}
```

The :MASK:OPERate command runs or stops the Mask Test function.

Query

```
:MASK:OPERate?
```

The :MASK:OPERate? query returns whether the Mask Test function is running or stopped.

Returned Format

```
{RUN | STOP}<NL>
```

OUTPut

Command

```
:MASK:OUTPut {FAIL | FAIL_SOUND | PASS | PASS_SOUND}
```

The :MASK:OUTPut command specifies the condition that, when detected, will cause an indication and whether the indication will include an audible beep.

Query

```
:MASK:OUTPut?
```

The :MASK:OUTPut? query returns the current output setting.

Returned Format

```
{FAIL | FAIL_SOUND | PASS | PASS_SOUND}<NL>
```

SOURce

Command

```
:MASK:SOURce {CHAN1 | CHAN2}
```

The :MASK:SOURce command selects either channel 1 or channel 2 as the source for the Mask Test.

Query

```
:MASK:SOURce?
```

The :MASK:SOURce? query returns the channel that is currently selected as the source for the Mask Test.

Returned Format

```
{CHAN1 | CHAN2}<NL>
```

STOPonoutput

Command

```
:MASK:STOPonoutput {{1 | ON} | {0 | OFF}}
```

The :MASK:STOPonoutput command specified whether the Mask Test stops when the output condition occurs.

Query

```
:MASK:STOPonoutput?
```

The :MASK:STOPonoutput? query returns the state of the “stop on output” control.

Returned Format

```
{1 | 0}<NL>
```

X

Command `:MASK:X <value>`

The :MASK:X command sets the mask's horizontal failure margin.

`<value>` 0.4 div to 4 div

Query `:MASK:X?`

The :MASK:X? query returns the current horizontal failure margin setting.

Returned Format `<value><NL>`

`<value> ::= 0.4 div to 4 div`

Y

Command

```
:MASK:Y <value>
```

The :MASK:Y command sets the mask's vertical failure margin.

`<value>` 0.4 div to 4 div

Query

```
:MASK:Y?
```

The :MASK:Y? query returns the current vertical failure margin setting.

Returned Format

```
<value><NL>
<value> ::= 0.4 div to 4 div
```

13

MEASure Commands

MEASure Commands

The commands in the MEASure subsystem are used to make parametric measurements on displayed waveforms.

These MEASure commands and queries are implemented in the 3000 Series oscilloscopes.

- "CLEar" on page 91.
- "FALLtime" on page 92.
- "FREQuency" on page 93.
- "NDUTycycle" on page 94.
- "NWIDth" on page 95.
- "OVERshoot" on page 96.
- "PDUTycycle" on page 97.
- "PERiod" on page 98.
- "PREShoot" on page 99.
- "PWIDth" on page 100.
- "RISetime" on page 101.
- "VAMPlitude" on page 102.
- "VAVerage" on page 103.
- "VBASe" on page 104.
- "VMAX" on page 105.
- "VMIN" on page 106.
- "VPP" on page 107.
- "VRMS" on page 108.
- "VTOP" on page 109.

CLEar

Command

```
:MEASure:CLEar
```

The :MEASure:CLEar command clears the on-screen measurement results.

FALLtime

Command

```
:MEASure:FALLtime [{CHANnel1 | CHANnel2}]
```

The :MEASure:FALLtime command displays the on-screen fall time measurement.

Query

```
:MEASure:FALLtime? [{CHANnel1 | CHANnel2}]
```

The :MEASure:FALLtime? query returns the fall time.

Returned Format

```
<value><NL>
```

`<value>` Time from the upper threshold time to the lower threshold time.

Note: the value returned can contain a "<" character, so it is best to read this value as a string.

FREQuency

Command `:MEASure:FREQuency [{CHANnel1 | CHANnel2}]`

The :MEASure:FREQuency command displays the on-screen ffrequency measurement.

Query `:MEASure:FREQuency? [{CHANnel1 | CHANnel2}]`

The :MEASure:FREQuency? query returns the measured frequency.

Returned Format `<value><NL>`

`<value>` The frequency value in Hertz of the first complete cycle on the screen using the mid-threshold levels of the waveform (in NR3 format).

NDUTycycle

Command `:MEASure:NDUTycycle [{CHANnel1 | CHANnel2}]`

The :MEASure:NDUTycycle command displays the on-screen negative duty cycle measurement.

Query `:MEASure:NDUTYcycle? [{CHANnel1 | CHANnel2}]`

The :MEASure:NDUTycycle? query returns the measured negative duty cycle in percent (%).

Returned Format `<value><NL>`

`<value>` The ratio (%) of the negative pulse width to the period.

Note: the value returned contains a "%" character, so read it as a string.

NWIDth

Command `:MEASure:NWIDth [{CHANnel1 | CHANnel2}]`

The :MEASure:NWIDth command displays the on-screen negative pulse width measurement.

Query `:MEASure:NWIDth? [{CHANnel1 | CHANnel2}]`

The :MEASure:NWIDth? query returns the measured width of the first negative pulse.

Returned Format `<value><NL>`

`<value>` The width of the first negative pulse on the screen using the mid-threshold levels of the waveform (in NR3 format).

OVERshoot

Command `:MEASure:OVERshoot [{CHANnel1 | CHANnel2}]`

The :MEASure:OVERshoot command displays the on-screen overshoot measurement.

Query `:MEASure:OVERshoot? [{CHANnel1 | CHANnel2}]`

The :MEASure:OVERshoot? query returns the measured overshoot.

Returned Format `<value><NL>`

`<value>` Ratio of overshoot to amplitude, in percent.
Note: the value returned contains a "%" character, so read it as a string.

PDUTycycle

Command `:MEASure:PDUTycycle [{CHANnel1 | CHANnel2}]`

The :MEASure:PDUTycycle command displays the on-screen positive duty cycle measurement.

Query `:MEASure:PDUTYcycle? [{CHANnel1 | CHANnel2}]`

The :MEASure:PDUTycycle? query returns the measured positive duty cycle in percent (%).

Returned Format `<value><NL>`

`<value>` The ratio (%) of the positive pulse width to the period.

Note: the value returned contains a "%" character, so read it as a string.

PERiod

Command `:MEASure:PERiod [{CHANnel1 | CHANnel2}]`

The :MEASure:PERiod command displays the on-screen period measurement.

Query `:MEASure:PERiod? [{CHANnel1 | CHANnel2}]`

The :MEASure:PERiod? query returns the measured period.

Returned Format `<value><NL>`

`<value>` Period of the first complete cycle on the screen (in NR3 format).

PREShoot

Command `:MEASure:PREShoot [{CHANnel1 | CHANnel2}]`

The :MEASure:PREShoot command displays the on-screen preshoot measurement.

Query `:MEASure:PREShoot? [{CHANnel1 | CHANnel2}]`

The :MEASure:PREShoot? query returns the measured preshoot.

Returned Format `<value><NL>`

`<value>` Ratio of preshoot to amplitude, in percent.
Note: the value returned contains a "%" character, so read it as a string.

PWIDth

Command `:MEASure:PWIDth [{CHANnel1 | CHANnel2}]`

The :MEASure:PWIDth command displays the on-screen positive pulse width measurement.

Query `:MEASure:PWIDth? [{CHANnel1 | CHANnel2}]`

The :MEASure:PWIDth? query returns the measured width of the first positive pulse.

Returned Format `<value><NL>`

`<value>` Width of the first positive pulse on the screen in seconds (in NR3 format).

RISetime

Command `:MEASure:RISetime [{CHANnel1 | CHANnel2}]`

The :MEASure:RISetime command displays the on-screen rise time measurement.

Query `:MEASure:RISetime? [{CHANnel1 | CHANnel2}]`

The :MEASure:RISetime? query returns the rise time.

Returned Format `<value><NL>`

`<value>` Rise time in seconds.

Note: the value returned can contain a "<" character, so it is best to read this value as a string.

VAMPlitude

Command `:MEASure:VAMPlitude [{CHANnel1 | CHANnel2}]`

The :MEASure:VAMPlitude command displays the on-screen voltage amplitude measurement.

Query `:MEASure:VAMPlitude? [{CHANnel1 | CHANnel2}]`

The :MEASure:VAMPlitude? query returns the calculated difference between the top and base voltage.

Returned Format `<value><NL>`

`<value>` Calculated difference between the top and base voltage (in NR3 format).

VAVerage

Command `:MEASure:VAVerage [{CHANnel1 | CHANnel2}]`

The :MEASure:VAVerage command displays the on-screen average voltage measurement.

Query `:MEASure:VAVerage? [{CHANnel1 | CHANnel2}]`

The :MEASure:VAVerage? query returns the calculated average voltage.

Returned Format `<value><NL>`

`<value>` The calculated average voltage (in NR3 format).

VBASe

Command `:MEASure:VBASe [{CHANnel1 | CHANnel2}]`

The :MEASure:VBASe command displays the on-screen base voltage measurement.

Query `:MEASure:VBASe? [{CHANnel1 | CHANnel2}]`

The :MEASure:VBASe? query returns the measured voltage value at the base.

Returned Format `<value><NL>`

`<value>` Voltage at the base of the waveform (in NR3 format).

VMAX

Command `:MEASure:VMAX [{CHANnel1 | CHANnel2}]`

The :MEASure:VMAX command displays the on-screen maximum voltage measurement.

Query `:MEASure:VMAX? [{CHANnel1 | CHANnel2}]`

The :MEASure:VMAX? query returns the measured absolute maximum voltage.

Returned Format `<value><NL>`

`<value>` Absolute maximum voltage present on the waveform (in NR3 format).

VMIN

Command

```
:MEASure:VMIN [{CHANnel1 | CHANnel2}]
```

The :MEASure:VMIN command displays the on-screen minimum voltage measurement.

Query

```
:MEASure:VMIN? [{CHANnel1 | CHANnel2}]
```

The :MEASure:VMIN? query returns the measured absolute minimum voltage.

Returned Format

```
<value><NL>
```

`<value>` Absolute minimum voltage present on the waveform (in NR3 format).

VPP

Command `:MEASure:VPP [{CHANnel1 | CHANnel2}]`

The :MEASure:VPP command displays the on-screen peak-to-peak voltage measurement.

Query `:MEASure:VPP? [{CHANnel1 | CHANnel2}]`

The :MEASure:VPP? query returns the peak-to-peak voltage.

Returned Format `<value><NL>`

`<value>` Peak-to-peak voltage (in NR3 format).

VRMS

Command `:MEASure:VRMS [{CHANnel1 | CHANnel2}]`

The :MEASure:VRMS command displays the on-screen RMS voltage measurement.

Query `:MEASure:VRMS? [{CHANnel1 | CHANnel2}]`

The :MEASure:VRMS? query returns the RMS voltage.

Returned Format `<value><NL>`

`<value>` RMS voltage of the selected waveform (in NR3 format).

VTOP

Command `:MEASure:VTOP [{CHANnel1 | CHANnel2}]`

The :MEASure:VTOP command displays the on-screen voltage at the top measurement.

Query `:MEASure:VTOP? [{CHANnel1 | CHANnel2}]`

The :MEASure:VTOP? query returns the measured voltage at the top.

Returned Format `<value><NL>`

`<value>` Voltage at the top of the waveform (in NR3 format).

14

SAVerecall Commands

SAVerecall Commands

The SAVerecall subsystem commands perform the setup and waveform storage operations. This allows saving and loading of waveforms and setups.

These SAVerecall commands and queries are implemented in the 3000 Series oscilloscopes:

- "LOAD" on page 113.
- "LOCation" on page 114.
- "SAVE" on page 115.
- "TYPE" on page 116.

LOAD

Command

```
:SAVerecall:LOAD
```

The :SAVerecall:LOAD command restores a setup or a waveform from the storage area defined by the :SAVerecall:LOCation command. The :SAVerecall:TYPE command determines if a waveform or setup is loaded.

LOCation

Command

```
:SAVerecall:LOCation {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
| 9 | 10}
```

The :SAVerecall:LOCation command defines which storage location is used by the :SAVerecall:LOAD and :SAVerecall:SAVE commands.

Query

```
:SAVerecall:LOCation?
```

The :SAVerecall:LOCation? query returns the currently selected storage location.

Returned Format

```
{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10}<NL>
```

SAVE

Command

```
:SAVerecall:SAVE
```

The :SAVerecall:SAVE command saves a setup or a waveform to a storage area. The :SAVerecall:LOCation command determines which storage area is used. The :SAVerecall:TYPE command determines if a waveform or setup is saved.

TYPE

Command

```
:SAVerecall:TYPE {WAVeforms | SETups}
```

The :SAVerecall:TYPE command defines whether a waveform or setup is stored in the storage location.

Query

```
:SAVerecall:TYPE?
```

The :SAVerecall:TYPE? query returns the currently selected storage type.

Returned Format

```
{WAVeforms | SETups}<NL>
```

15

TIMebase Commands

TIMebase Commands

The TIMebase subsystem commands control the horizontal (X axis) oscilloscope functions. These TIMebase commands and queries are implemented in the oscilloscope:

- "DELayed" on page 119.
- "HOLDoff" on page 120.
- "POSition" on page 121.
- "SCALe" on page 122.
- "SCAN" on page 123.

DELayed

Command

```
:TIMebase:DELayed {{1 | ON} | {0 | OFF}}
```

The :TIMebase:DELayed command enables or disables the Delayed Sweep mode.

Query

```
:TIMebase:DELayed?
```

The :TIMebase:DELayed? query returns the state of the Delayed Sweep mode control.

Returned Format

```
{1 | 0}<NL>
```

HOLDoff

Command `:TIMebase:HOLDoff <holdoff_time>`

The :TIMebase:HOLDoff command sets the holdoff time.

The holdoff time is the oscilloscope's waiting period before starting a new trigger. During the holdoff time oscilloscope will not trigger until the holdoff has expired. Holdoff can be used to stabilize a waveform.

`<holdoff_time>` 100 ns to 1.5 s.

Query `:TIMebase:HOLDoff?`

The :TIMebase:HOLDoff? query returns the current holdoff time value.

Returned Format `<holdoff_time><NL>`

`<holdoff_time> ::= in NR3 format`

POSition

Command `:TIMebase:POSition <delay_time>`

The :TIMebase:POSition sets the amount of time from the center of screen to the trigger point of the waveform.

`<delay_time>` If the horizontal time base is set between 50 s/div and 50 ms/div, the delayed trigger time range is:

$$\text{delay\_time} = \pm 6 div \times \text{time base setting}$$

If the horizontal time base is set to less than 50 ms/div then the delayed trigger time range is:

$$\text{delay\_time} = -14 div \times \text{time base setting to 1s}$$

Query `:TIMebase:POSition?`

The :TIMebase:POSition? query returns the value of the delayed trigger.

Returned Format `<delay_time><NL>`

SCALe

Command `:TIMebase:SCALe <time>`

The :TIMebase:SCALe command sets the time base scale. This corresponds to the horizontal scale value displayed as Time on the oscilloscope screen.

`<time>` The time value is in a 1-2-5 sequence (for example, 1.0E-9, 2.0E-9, 5.0E-9, ..., 1.0E+00, 2.0E+00, 5.0E+00) from:

- 1 ns/div to 50 s/div (DSO3202A).
- 2 ns/div to 50 s/div (DSO3152A and DSO3102A).
- 5 ns/div to 50 s/div (DSO3062A).

Query `:TIMebase:SCALe?`

The :TIMebase:SCALe? query returns the current horizontal time setting.

Returned Format `<time><NL>`

SCAN

Query

```
:TIMebase:SCAN?
```

The :TIMebase:SCAN? query returns whether the oscilloscope is in the Roll Mode (1 = yes, 0 = no).

Returned Format

```
{1 | 0}<NL>
```

16

TRIGger Commands

TRIGger Commands

The oscilloscope trigger circuitry helps you locate the waveform you want to view. Edge triggering identifies a trigger condition by looking for the slope and voltage level (trigger level) on the source you select.

The commands in the TRIGger subsystem define the conditions for triggering. The command set has been defined to closely represent the front-panel trigger menus.

These TRIGger commands and queries are implemented in the 3000 Series oscilloscopes:

- "[:EDGE]:COUPling" on page 127.
- "[:EDGE]:LEVel" on page 128.
- "[:EDGE]:SLOPe" on page 129.
- "[:EDGE]:SOURe" on page 130.
- "[:EDGE]:SWEep" on page 131.
- "MODE" on page 132.
- "PULSe:MODE" on page 133.
- "PULSe:WIDTh" on page 134.
- "STATus" on page 135.
- "VIDeo:POLarity" on page 136.
- "VIDeo:STANdard" on page 137.
- "VIDeo:SYNC" on page 138.

[:EDGE]:COUPling

Command

```
:TRIGger[:EDGE]:COUPling {DC | AC | HF | LF}
```

The :TRIGger[:EDGE]:COUPling command sets the input coupling for the selected trigger sources. The coupling can be set to DC, AC, HF, or LF.

- DC sets the input coupling to DC.
- AC sets the input coupling to AC (50 Hz cutoff).
- LF sets the input coupling to low frequency reject (100 kHz cutoff).
- HF sets the input coupling to high frequency reject (10 kHz cutoff).

Query

```
:TRIGger[:EDGE]:COUPling?
```

The query returns the currently selected edge coupling.

Returned Format

```
{DC | AC | HF | LF}<NL>
```

[:EDGE]:LEVel

Command `:TRIGger[:EDGE]:LEVel <level>}`

The :TRIGger[:EDGE]:LEVel command specifies the trigger level.

`<level>` A number in the range of +12div to -12div.

Query `:TRIGger[:EDGE]:LEVel?`

The query returns the trigger level.

Returned Format `<level><NL>`

[:EDGE]:SLOPe

Command

```
:TRIGger[:EDGE]:SLOPe {NEGative | POSitive}
```

The :TRIGger[:EDGE]:SLOPe command specifies the slope of the edge used to trigger the oscilloscope.

Query

```
:TRIGger[:EDGE]:SLOPe?
```

The query returns the currently selected edge slope.

Returned Format

```
{NEGative | POSitive}<NL>
```

[:EDGE]:SOURe

Command

```
:TRIGger[:EDGE]:SOURe {CHANnel1 | CHANnel2 | EXT |EXT5
| ACLine}
```

The :TRIGger[:EDGE]:SOURe command sets the source used for triggering.

Query

```
:TRIGger[:EDGE]:SOURe?
```

The query returns the currently selected trigger source.

Returned Format

```
{CHANnel1 | CHANnel2 | EXT | EXT5 | ACLine}<NL>
```

[:EDGE]:SWEep

Command `:TRIGger[:EDGE]:SWEep {AUTO | NORMal}`

The :TRIGger[:EDGE]:SWEep command selects the oscilloscope sweep mode.

`<AUTO>` When you select AUTO, if a trigger event does not occur within a time determined by the oscilloscope settings, the oscilloscope automatically forces a trigger which causes the oscilloscope to sweep. If the frequency of your waveform is 20 Hz or less, you should not use the AUTO sweep mode because it is possible that the oscilloscope will automatically trigger before your waveform trigger occurs.

`<NORMal>` When you select MORMal, if no trigger occurs, the oscilloscope will not sweep, and no waveform data will appear on the screen.

Query `:TRIGger[:EDGE]:SWEep?`

The query returns the specified channel's sweep mode.

Returned Format `[:TRIGger:SWEep] {AUTO | NORMal}<NL>`

MODE

Command

```
:TRIGger:MODE {EDGE | PULSe | TV}
```

The :TRIGger:MODE command sets the trigger mode.

Query

```
:TRIGger:MODE?
```

The query returns the currently selected trigger mode.

Returned Format

```
{EDGE | PULSE | TV}<NL>
```

PULSe:MODE

Command

```
:TRIGger:PULSe:MODE {+GREaterthan | +LESSthan |
+EQUal | -GREaterthan | -LESSthan | -EQUal}
```

The :TRIGger:PULSe:MODE command sets the pulse trigger mode. The "+" options are for positive pulses; the "-" options are for negative pulses.

Query

```
:TRIGger:PULSe:MODE?
```

The query returns the currently selected pulse trigger mode.

Returned Format

```
{+GREaterthan | +LESSthan | +EQUal | -GREaterthan | -LESSthan |
-EQUal}<NL>
```

PULSe:WIDTh

Command

```
:TRIGger:PULSe:WIDTh <width>}
```

The :TRIGger:PULSe:WIDTh command specifies the pulse trigger width.

`<width>` 20 ns to 10 s.

Query

```
:TRIGger:PULSe:WIDTh?
```

The query returns the current pulse trigger width setting.

Returned Format

```
<width><NL>
<width> ::= in NR3 format
```

STATus

Query

```
:TRIGger:STATus?
```

The query returns the current trigger status.

Returned Format

```
{STOP | T'D | WAIT}<NL>
```

VIDeo:POLarity

Command

```
:TRIGger:VIDeo:POLarity {POSitive | NEGative}
```

The :TRIGger:VIDeo:POLarity command sets the edge of the sync pulse to trigger on.

Query

```
:TRIGger:VIDeo:POLarity?
```

The query returns the current sync pulse edge setting.

Returned Format

```
{POSitive | NEGative}<NL>
```

VIDeo:STANdard

Command

```
:TRIGger:VIDeo:STANdard {NTSC | PALSecam}
```

The :TRIGger:VIDeo:STANdard command sets they type of video waveform to trigger on.

Query

```
:TRIGger:VIDeo:STANdard?
```

The query returns the currently selected video trigger waveform type.

Returned Format

```
{NTSC | PALSecam}<NL>
```

VIDeo:SYNC

Command

```
:TRIGger:VIDeo:SYNC {FIELd1 | FIELd2 | LINE |
ALLLines}
```

The :TRIGger:VIDeo:SYNC command sets the line or field in the video waveform to trigger on.

- FIELd1 triggers on an odd field.
- PIELd2 triggers on and even field.
- LINE triggers on a selected line.
- ALLLines triggers on all lines.

Query

```
:TRIGger:VIDeo:SYNC?
```

The query returns the current video trigger line/field setting.

Returned Format

```
{FIELd1 | FIELd2 | LINE | ALLLines}<NL>
```

17

WAVeform Commands

WAVeform Commands

The WAVeform subsystem is used to transfer waveform data from the oscilloscope to a computer. It contains commands to transfer waveform information and waveform data from the oscilloscope.

These WAVeform commands and queries are implemented in the 3000 Series oscilloscopes:

- "DATA?" on page 141.
- "ERASeofroll?" on page 142.
- "MAXPeakdetect?" on page 143.
- "MEMorydata?" on page 144.
- "MINPeakdetect?" on page 145.
- "SCREENDATA?" on page 146.
- "SCREENMAX?" on page 147.
- "SCREENMIN?" on page 148.
- "SOURce" on page 149.
- "STARtofroll?" on page 150.
- "SYSMemsize?" on page 151.
- "TPOSition?" on page 152.
- "WINDowzoom?" on page 153.
- "WINMemsize?" on page 154.
- "WPOSition?" on page 155.
- "XEND?" on page 156.
- "XINCrement?" on page 157.
- "XORigin?" on page 158.
- "XSTart?" on page 159.
- "YINCrement?" on page 160.
- "YORigin?" on page 161.

DATA?

Query

```
:WAVeform:DATA?
```

The :WAVeform:DATA? query outputs waveform data to the computer over the selected interface. This query is the same as the :WAVeform:SCREENDATA? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

“Block Data” on page 17.

ERASeofroll?

Query

```
:WAVeform:ERASeofroll?
```

The :WAVeform:ERASeofroll? query returns the width in points of the erase band when in Roll Mode.

Returned Format

```
<roll_erase_wid><NL>
```

```
<roll_erase_wid> ::= integer in NR1 format.
```

MAXPeakdetect?

Query

```
:WAVeform:MAXPeakdetect?
```

The :WAVeform:MAXPeakdetect? query outputs the peak detect maximum waveform data to the computer over the selected interface. This query is the same as the :WAVeform:SCREENMAX? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

“Block Data” on page 17.

MEMorydata?

Query

```
:WAVeform:MEMorydata?
```

The :WAVeform:MEMorydata? query outputs the reference memory waveform data to the computer over the selected interface.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

"Block Data" on page 17.

MINPeakdetect?

Query

```
:WAVeform:MINPeakdetect?
```

The :WAVeform:MINPeakdetect? query outputs the peak detect maximum waveform data to the computer over the selected interface. This query is the same as the :WAVeform:SCREENMIN? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

"Block Data" on page 17.

SCREENDATA?

Query

```
:WAVeform:SCREENDATA?
```

The :WAVeform:SCREENDATA? query outputs waveform data to the computer over the selected interface. This query is the same as the :WAVeform:DATA? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

"Block Data" on page 17.

SCREENMAX?

Query

```
:WAVeform:SCREENMAX?
```

The :WAVeform:SCREENMAX? query outputs the peak detect maximum waveform data to the computer over the selected interface. This query is the same as the :WAVeform:MAXPeakdetect? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

"Block Data" on page 17.

SCREENMIN?

Query

```
:WAVeform:SCREENMIN?
```

The :WAVeform:SCREENMIN? query outputs the peak detect maximum waveform data to the computer over the selected interface. This query is the same as the :WAVeform:MINPeakdetect? query.

Use the :WAVeform:SOURce command to determine which channel waveform data is output.

Returned Format

```
<block_data><NL>
```

See Also

"Block Data" on page 17.

SOURce

Command

```
:WAVeform:SOURce {CHANnel1 | CHANnel2}
```

The :WAVeform:SOURce command selects a channel as the waveform source.

Query

```
:WAVeform:SOURce?
```

The :WAVeform:SOURce? query returns the currently selected waveform source.

Returned Format

```
{CHANnel1 | CHANnel2}<NL>
```

STARtofroll?

Query

```
:WAVeform:STARtofroll?
```

The :WAVeform:STARtofroll? query returns the waveform's starting position when in Roll Mode.

Returned Format

```
<roll_start_pos><NL>
```

```
<roll_Start_pos> ::= Integer in NR1 format.
```

SYSMemsize?

Query

```
:WAVeform:SYSMemsize?
```

The :WAVeform:SYSMemsize? query returns the points value of the waveform in system memory.

Returned Format

```
<memory_data_size><NL>
```

```
<memory_data_size> ::= Integer in NR1 format.
```

See Also

“MEMorydata?” on page 144.

TPOSition?

Query

```
:WAVeform:TPOSition?
```

The :WAVeform:TPOSition? query returns the points value in the current waveform preamble.

The points value is the number of time buckets contained in the waveform selected with the :WAVeform:SOURce command.

Returned Format

```
<trig_pos><NL>
```

```
<trig_pos> ::= Integer in NR1 format.
```

WINDowzoom?

Query

```
:WAVeform:WINDowzoom?
```

The :WAVeform:WINDowzoom? query returns zzz ???.

Returned Format

```
<window_zoom><NL>
<window_zoom> ::= Integer in NR1 format.
```

WINMemsize?

Query

```
:WAVeform:WINMemsize?
```

The :WAVeform:WINMemsize? query returns the points value of the waveform in the window's memory.

Returned Format

```
<window_data_size><NL>
```

```
<window_data_size> ::= Integer in NR1 format.
```

See Also

"DATA?" on page 141.

WPOSition?

Query

```
:WAVeform:WPOSition?
```

The :WAVeform:WPOSition? query returns the points value in the current waveform position.

Returned Format

```
<wave_pos><NL>
```

```
<wave_pos> ::= Integer in NR1 format.
```

XEND?

Query

```
:WAVeform:XEND?
```

The :WAVeform:XEND? query returns the horizontal end position points value.

Returned Format

```
<screen_wave_endx><NL>
<screen_wave_endx> ::= Integer in NR1 format.
```

XINCrement?

Query

```
:WAVeform:XINCrement?
```

The :WAVeform:XINCrement? query returns the time difference between consecutive data points for the currently specified waveform source.

- For time domain waveforms, this is the time difference between consecutive data points.
- For VERSus type waveforms, this is the duration between levels on the X axis.
- For voltage waveforms, this is the voltage corresponding to one level.

Returned Format

```
<value><NL>
```

```
<value> ::= A real number representing the time between data
points on the X axis.
```

XORigin?

Query

```
:WAVeform:XORigin?
```

The :WAVeform:XORigin? query returns the X-axis value of the first data point in the data record.

- For time domain waveforms, it is the time of the first point.
- For VERSus type waveforms, it is the X-axis value at level zero.
- For voltage waveforms, it is the voltage at level zero.

The value returned by this query is treated as a double precision 64-bit floating point number.

Returned Format

```
<value><NL>

<value> ::= A real number representing the X-axis value of
the first data point in the data record.
```

XSTart?

Query

```
:WAVeform:XSTart?
```

The :WAVeform:XSTart? query returns the horizontal start position points value.

Returned Format

```
<screen_wave_startx><NL>
<screen_wave_startx> ::= Integer in NR1 format
```

YINCrement?

Query

```
:WAVeform:YINCrement?
```

The :WAVeform:YINCrement? query returns the y-increment voltage value for the currently specified source.

This voltage value is the voltage difference between two adjacent waveform data digital codes.

Adjacent digital codes are codes that differ by one least significant bit. For example, the digital codes 24680 and 24681 vary by one least significant bit.

Returned Format

```
<real_value><NL>

<real_value> ::= A real number in exponential format.
```

YORigin?

Query

```
:WAVeform:YORigin?
```

The :WAVeform:YORigin? query returns the y-origin voltage value for the currently specified source. The voltage value returned is the voltage value represented by the waveform data digital code 00000.

Returned Format

```
<real_value><NL>
```

```
<real_value> ::= A real number in exponential format.
```

Index

Symbols

...
    Ellipsis 16

A

ACQuire Commands 42
    AVERages 43
    MODE 44
    SRATe 45
    TYPE 46
ACQUIRE, KEY command 75
ASCII
    character 32 15
    linefeed 16
:AUTO, root level command 35
AUTO_SCALE, KEY command 75
AVERages, ACQuire command 43

B

basic operations 14
BEEP Commands 48
    ENABLe 49
block data 17
Braces 16
Brackets, Square 16
BWLimit, CHANnel<n> command 53

C

CH1, KEY command 75
CH1_POS_DEC, KEY command 75
CH1_POS_INC, KEY command 75
CH1_SCALE_DEC, KEY command 75
CH1_SCALE_INC, KEY command 75
CH2, KEY command 75
CH2_POS_DEC, KEY command 75
CH2_POS_INC, KEY command 75
CH2_SCALE_DEC, KEY command 75
CH2_SCALE_INC, KEY command 75
CHANnel<n> Commands 52
    BWLimit 53
    COUPling 54
    DISPlay 55
    INVert 56
    OFFSet 57
    PROBe 58
    SCALe 59
cleaning the instrument 167
Clear Status (*CLS) 29
CLEar, DISPlay command 67
CLEar, MEASure command 91
*CLS (Clear Status) 29
Commands Quick Reference 20
Common Commands 28
    *CLS (Clear Status) 29
    *IDN? (Identification Number) 30
    *OPC (Operation Complete) 31
    *RST (Reset) 32
COUNter Commands 62
    ENABle 63
    VALue 64
COUPling, CHANnel<n> command 54
COUPling, EDGE TRIGger command 127
CURSOR, KEY command 75

D

DATA, WAVeform command 141
decimal 32 (ASCII space) 15
DELayed, TIMebase command 119
DISPlay Commands 66
    CLEar 67
    GRID 68
    MENUdisplay 69
    PERSistence 70
    SCReen 71
    TYPE 72
display persistence 70
DISPlay, CHANnel<n> command 55
DISPLAY, KEY command 75

E

[:EDGE]:COUPling, TRIGger command 127
[:EDGE]:LEVel, TRIGger command 128
[:EDGE]:SLOPe, TRIGger command 129
[:EDGE]:SOURce, TRIGger command 130
[:EDGE]:SWEep, TRIGger command 131
Ellipsis ... 16
embedded strings 14
ENABle, BEEP command 49
ENABle, COUNter command 63
ENABle, MASK command 81
End Of String (EOS) 16
End Of Text (EOT) 16
End-Or-Identify (EOI) 16
ERASeofroll, WAVeform command 142

F

F1, KEY command 75
F2, KEY command 75
F3, KEY command 75
F4, KEY command 75
F5, KEY command 75
FALLtime, MEASure command 92
FORCE, KEY command 75
:ForceTrig, root level command 36
FREQuency, MEASure command 93

G

GPIB configuration 12
GRID, DISPlay command 68

H

headers 14
HOLDoff, TIMebase command 120
horizontal functions, controlling 118

I

I/O module
    GPIB configuration 12
    Installation 8
    RS-232 configuration 10
Identification Number (*IDN?) 30
*IDN? (Identification Number) 30
instruction headers 14
Introduction to Programming 14
INVert, CHANnel<n> command 56

K

KEY Commands 74
    ACQUIRE 75
    AUTO_SCALE 75
    CH1 75
    CH1_POS_DEC 75
    CH1_POS_INC 75
    CH1_SCALE_DEC 75
    CH1_SCALE_INC 75
    CH2 75
    CH2_POS_DEC 75
    CH2_POS_INC 75
    CH2_SCALE_DEC 75

CH2_SCALE_INC 75
CURSOR 75
DISPLAY 75
F1 75
F2 75
F3 75
F4 75
F5 75
FORCE 75
LOCK 77
MAIN_DELAYED 75
MATH 76
MEASURE 76
MNU_ON_OFF 76
MODE_COUPLING 76
PROMPT_TIME 76
REF 76
RUN 76
SAVE_RECALL 76
SINGLE 76
TIME_POS_DEC 76
TIME_POS_INC 76
TIME_SCALE_DEC 76
TIME_SCALE_INC 76
TRIG%50 76
TRIG_LVL_DEC 76
TRIG_LVL_INC 76
UTILITY 76

L

LEVel, EDGE TRIGger command 128
linefeed 16
LOAD, SAVerecall command 113
LOCation, SAVerecall command 114
LOCK, KEY command 77

M

MAIN_DELAYED, KEY command 75
MASK Commands 80
ENABle 81
OPERate 82
OUTPut 83
SOURce 84
STOPonoutput 85
X 86
Y 87
MATH, KEY command 76
MAXPeakdetect, WAVeform command 143
MEASure Commands 90
CLEar 91
FALLtime 92
FREQuency 93
NWIDth 95
OVERshoot 96
PDUTycycle 94, 97
PERiod 98
PREShoot 99
PWIDth 100
RISetime 101
VAMPlitude 102
VAVerage 103
VBASe 104
VMAX 105
VMIN 106
VPP 107
VRMS 108
VTOP 109
MEASURE, KEY command 76
MEMorydata, WAVeform command 144
MENUdisplay, DISPlay command 69
MINPeakdetect, WAVeform command 145
Mnemonic Truncation 15
MNU_ON_OFF, KEY command 76
MODE, ACQuire command 44
MODE, PULSe TRIGger command 133
MODE, TRIGger command 132
MODE_COUPLING, KEY command 76

N

NL (New Line) 16
NWIDth, MEASure command 95

O

OFFSet, CHANnel<n> command 57
*OPC (Operation Complete) 31
OPERate, MASK command 82
Operation Complete (*OPC?) 31
OUTPut, MASK command 83
OVERshoot, MEASure command 96

P

PDUTycycle, MEASure command 94, 97
PERiod, MEASure command 98
PERSistence, DISPlay command 70
POLarity, VIDeo TRIGger command 136
POSition, TIMebase command 121
PREShoot, MEASure command 99
PROBe, CHANnel<n> command 58
program message terminator 16
programming basics 14
PROMPT_TIME, KEY command 76
PULSe:MODE, TRIGger command 133
PULSe:WIDTh, TRIGger command 134
PWIDth, MEASure command 100

Q

Query 14
quick reference, commands 20

R

REF, KEY command 76
Remote Command Tips 17
remote programming basics 14
Reset (*RST) 32
RISetime, MEASure command 101
Root Level Commands 34
:AUTO 35
:ForceTrig 36
:RUN 37
:STOP 38
:Trig%50 39
RS-232 configuration 10
*RST (Reset) 32
rule of truncation 15
:RUN, root level command 37
RUN, KEY command 76

S

SAVE, SAVerecall command 115
SAVE_RECALL, KEY command 76
SAVerecall Commands 112
LOAD 113
LOCation 114
SAVE 115
TYPE 116
SCALe, CHANnel<n> command 59
SCALe, TIMebase command 122
SCAN, TIMebase command 123
SCReen, DISPlay command 71
SCREENDATA, WAVeform command 146
SCREENMAX, WAVeform command

147
SCREENMIN, WAVeform command 148
separator 15
serial prefix, reading 30
short-form mnemonics 15
SINGLE, KEY command 76
SLOPe, EDGE TRIGger command 129
software version, reading 30
SOURce, EDGE TRIGger command 130
SOURce, MASK command 84
SOURce, WAVeform command 149
Square Brackets 16
SRATe, ACQuire command 45
STANdard, VIDeo TRIGger command 137
STARtofroll, WAVeform command 150
STATus, TRIGger command 135
:STOP, root level command 38
STOPonoutput, MASK command 85
SWEep, EDGE TRIGger command 131
SYNC, VIDeo TRIGger command 138
SYSMemsize, WAVeform command 151

T

Terminator 16
TIME_POS_DEC, KEY command 76
TIME_POS_INC, KEY command 76
TIME_SCALE_DEC, KEY command 76
TIME_SCALE_INC, KEY command 76
TIMebase Commands 118
DELayed 119
HOLDoff 120
POSition 121
SCALe 122
SCAN 123
tips, remote command 17
TPOSition, WAVeform command 152
:Trig%50, root level command 39
TRIG%50, KEY command 76
TRIG_LVL_DEC, KEY command 76
TRIG_LVL_INC, KEY command 76
TRIGger Commands 126
[:EDGE]:COUPling 127
[:EDGE]:LEVel 128
[:EDGE]:SLOPe 129
[:EDGE]:SOURce 130
[:EDGE]:SWEep 131
MODE 132
PULSe:MODE 133
PULSe:WIDTh 134
SATus 135
VIDeo:POLarity 136
VIDeo:STANdard 137
VIDeo:SYNC 138
Truncation Rule 15
TYPE, ACQuire command 46
TYPE, DISPlay command 72
TYPE, SAVerecall command 116

U

UTILITY, KEY command 76

V

VALue, COUNter command 64
VAMPlitude, MEASure command 102
VAVerage, MEASure command 103
VBASe, MEASure command 104
VIDeo:POLarity, TRIGger command 136
VIDeo:STANdard, TRIGger command 137
VIDeo:SYNC, TRIGger command 138
VMAX, MEASure command 105
VMIN, MEASure command 106
VPP, MEASure command 107
VRMS, MEASure command 108
VTOP, MEASure command 109

W

WAVeform Commands 140
DATA 141
ERASeofroll 142
MAXPeakdetect 143
MEMorydata 144
MINPeakdetect 145
SCREENDATA 146
SCREENMAX 147
SCREENMIN 148
SOURce 149
STARtofroll 150
SYSMemsize 151
TPOSition 152
WINDowzoom 153
WINMemsize 154
WPOSition 155
XEND 156
XINCrement 157
XORigin 158
XSTart 159
YINCrement 160
YORigin 161
white space (separator) 15
WIDTh, PULSe TRIGger command 134
WINDowzoom, WAVeform command 153
WINMemsize, WAVeform command 154
WPOSition, WAVeform command 155

X

x axis, controlling 118
X, MASK command 86
XEND, WAVeform command 156
XINCrement, WAVeform command 157
XORigin, WAVeform command 158
XSTart, WAVeform command 159

Y

Y, MASK command 87
YINCrement, WAVeform command 160
YORigin, WAVeform command 161

Safety Notices

This apparatus has been designed and tested in accordance with IEC Publication 1010, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

Warnings

• Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.

• Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock or fire hazard.

• If you energize this instrument by an auto transformer (for voltage reduction or mains isolation), the common terminal must be connected to the earth terminal of the power source.

• Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.

• Service instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

• Do not install substitute parts or perform any unauthorized modification to the instrument.

• Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.

• Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

• Do not use the instrument in a manner not specified by the manufacturer.

To clean the instrument

If the instrument requires cleaning: (1) Remove power from the instrument. (2) Clean the external surfaces of the instrument with a soft cloth dampened with a mixture of mild detergent and water. (3) Make sure that the instrument is completely dry before reconnecting it to a power source.

Safety Symbols

Instruction manual symbol: the product is marked with this symbol when it is necessary for you to refer to the instruction manual in order to protect against damage to the product..

Hazardous voltage symbol.

Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

Agilent Technologies Inc.
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901-2197, U.S.A.

Notices

Manual Part Number
D3000-97016, September 2008

Print History

D3000-97016, September 2008
D3000-97011, August 2006
D3000-97001, April 2005

Agilent Technologies, Inc.
1601 California Street
Palo Alto, CA 94304 USA

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Document Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

Trademark Acknowledgements

Windows and MS Windows are U.S. registered trademarks of Microsoft Corporation.