

# Programmer Manual



## AWG400/500/600 Series Arbitrary Waveform Generator

**070-A810-50**

This document supports program version 3.0 of  
AWG400 Series, version 4.0 of AWG500 Series  
and program version 4.0 of AWG600 Series.

**[www.tektronix.com](http://www.tektronix.com)**

Copyright © Tektronix Japan, Ltd. All rights reserved.

Copyright © Tektronix, Inc. All rights reserved.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supercedes that in all previously published material. Specifications and price change privileges reserved.

Tektronix Japan, Ltd., 5-9-31 Kitashinagawa, Shinagawa-ku, Tokyo 141-0001 Japan

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

**THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.**



# Table of Contents

<b>List of Tables</b> .....	<b>vi</b>
<b>List of Figures</b> .....	<b>viii</b>
<b>Preface</b> .....	<b>xi</b>

## Getting Started

<b>Getting Started</b> .....	<b>1-1</b>
Manual Overview .....	1-1
Setting Up Remote Communications Using GPIB .....	1-4
Setting Up Remote Communications Using Ethernet .....	1-8

## Syntax and Commands

<b>Command Syntax</b> .....	<b>2-1</b>
SCPI Commands and Queries .....	2-2
IEEE 488.2 Common Commands .....	2-9
Constructed Mnemonics .....	2-10
Syntax Diagrams .....	2-15
<b>Command Groups</b> .....	<b>2-17</b>
Functional Groups .....	2-17
Command Quick Reference .....	2-18
Command Summaries .....	2-20
<b>Command Descriptions</b> .....	<b>2-31</b>
ABORt (No Query Form) .....	2-31
ABSTouch (No Query Form) .....	2-32
AWGControl:CLOCK:SOURce (?) .....	
AWG400/500 series Only .....	2-37
AWGControl:DOUTput<x>[:STATE] (?) .....	2-37
AWGControl:ENHanced:SEquence[:JMODE] (?) .....	2-39
AWGControl:EVENT[:LOGic][:IMMEDIATE] (No Query Form) .....	2-39
AWGControl:EVENT:SOFTware[:IMMEDIATE] (No Query Form) .....	2-40
AWGControl:EVENT:TABLE[:IMMEDIATE] (No Query Form) .....	2-41
AWGControl:FG:FREQuency[:CW]:FIXed (?) .....	2-41
AWGControl:FG<n>:FUNctIon[:SHAPE] (?) .....	2-42
AWGControl:FG<n>:PHASe[:ADJusT] (?) .....	
AWG420, AWG430 and AWG520 Only .....	2-44
AWGControl:FG<n>:POLarity (?) .....	2-45
AWGControl:FG<n>:PULSe:DCYCLE (?) .....	2-46
AWGControl:FG[:STATE] (?) .....	2-47
AWGControl:FG<n>:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude] (?) .....	2-48
AWGControl:FG<n>:VOLTage[:LEVel][:IMMEDIATE]:OFFSet (?) .....	2-49
AWGControl:RMODE (?) .....	2-50
AWGControl:RStAtE? (Query Only) .....	2-51
AWGControl:RUN[:IMMEDIATE] (No Query Form) .....	2-51
AWGControl:SREStore (No Query Form) .....	2-52
AWGControl:SSAVe (No Query Form) .....	2-53
AWGControl:STOP[:IMMEDIATE] (No Query Form) .....	2-53

*CAL? (Query Only) . . . . .	2-54
CALibration[:ALL] (?) . . . . .	2-55
*CLS (No Query Form) . . . . .	2-56
DIAGnostic:DATA? (Query Only) . . . . .	2-56
DIAGnostic[:IMMediate] (?) . . . . .	2-57
DIAGnostic:SElect (?) . . . . .	2-58
DISPlay:BRIGhtness (?)	
AWG500/600 series Only . . . . .	2-59
DISPlay:ENABle (?)	
AWG400 series Only . . . . .	2-59
DISPlay:HILight:COLor (?)	
AWG400 series Only . . . . .	2-60
*ESE (?) . . . . .	2-61
*ESR? (Query Only) . . . . .	2-62
HCOPy:DESTination (No Query Form) . . . . .	2-62
HCOPy:DEvice:COLor (?)	
AWG400 series Only . . . . .	2-63
HCOPy:DEvice:LANGuage (?) . . . . .	2-63
HCOPy[:IMMediate] (No Query Form) . . . . .	2-64
HCOPy:SDUMp[:IMMediate] (No Query Form) . . . . .	2-65
*IDN? (Query Only) . . . . .	2-65
MMEMory:CATalog? (Query Only) . . . . .	2-66
MMEMory:CDIRectory (?) . . . . .	2-67
MMEMory:CLOSe (No Query Form) . . . . .	2-67
MMEMory:COpy (No Query Form) . . . . .	2-68
MMEMory:DATA (?) . . . . .	2-69
MMEMory:DELeTe (No Query Form) . . . . .	2-69
MMEMory:FEED (?) . . . . .	2-70
MMEMory:INITialize (No Query Form) . . . . .	2-71
MMEMory:MDIRectory (No Query Form) . . . . .	2-72
MMEMory:MOVE (No Query Form) . . . . .	2-72
MMEMory:MSIS (?) . . . . .	2-73
MMEMory:NAME (?) . . . . .	2-74
MMEMory:OPEN (No Query Form) . . . . .	2-75
*OPC (?) . . . . .	2-76
*OPT? (Query Only) . . . . .	2-76
OUTPut<x>:FILTer[:LPASS]:FREQuency (?) . . . . .	2-77
OUTPut<x>:ISTate (?)	
AWG400 series(except Opt05) , AWG510 and AWG610 Only . . . . .	2-78
OUTPut<x>[:STATe] (?) . . . . .	2-79
*PSC (?) . . . . .	2-80
*RST (No Query Form) . . . . .	2-81
[SOURce<x>]:COMBine:FEED (?)	
AWG400/500 series Only . . . . .	2-81
[SOURce<x>]:FREQuency[:CW FIXed] (?) . . . . .	2-83
[SOURce<x>]:FUNction:USER (?) . . . . .	2-84
[SOURce<x>]:MARKer[1 2]:DELay (?)	
AWG500/600 series Only . . . . .	2-85
[SOURce<x>]:MARKer[1 2]:VOLTagE[:LEVel][:IMMediate]:HIGH (?)	
AWG500/600 series Only . . . . .	2-86
[SOURce<x>]:MARKer[1 2]:VOLTagE[:LEVel][:IMMediate]:LOW (?)	
AWG500/600 series Only . . . . .	2-87

SOURce<x>:POWer[:LEVel][:IMMediate][:AMPLitude] (?)	
AWG400/500 series Only	2-88
[SOURce<x>]:ROSCillator:SOURce (?)	2-89
SOURce[2 3 5 6]:SKEW (?)	
AWG400 series Only	2-90
[SOURce<x>]:VOLTage[:LEVel][:IMMediate][:AMPLitude] (?)	2-90
SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH (?)	
AWG500 series Only	2-91
SOURce5:VOLTage[:LEVel][:IMMediate]:LOW (?)	
AWG500 series Only	2-92
[SOURce<x>]:VOLTage[:LEVel][:IMMediate]:OFFSet (?)	2-93
*SRE (?)	2-94
STATus:OPERation:CONDition? (Query Only)	2-95
STATus:OPERation:ENABle (?)	2-96
STATus:OPERation[:EVENT]? (Query Only)	2-96
STATus:PRESet (No Query Form)	2-97
STATus:QUESTionable:CONDition? (Query Only)	2-97
STATus:QUESTionable:ENABle (?)	2-98
STATus:QUESTionable[:EVENT]? (Query Only)	2-99
STATus:QUEue[:NEXT]? (Query Only)	
AWG500/600 series Only	2-99
*STB? (Query Only)	2-100
SYSTem:BEEPer[:IMMediate] (No Query Form)	2-101
SYSTem:COMMunicate:LAN:DHCP[:CLient]:LEASe:TIME (?)	
AWG400 series Only	2-101
SYSTem:COMMunicate:LAN:DHCP[:CLient][:STATe] (?)	
AWG400 series Only	2-102
SYSTem:COMMunicate:LAN:FTP[:SERVer][:STATe] (?)	2-103
SYSTem:COMMunicate:LAN:FTP[:SERVer]:VERSion (?)	2-103
SYSTem:COMMunicate:LAN:GATeway[1 2 3]:ADDRess (?)	2-104
SYSTem:COMMunicate:LAN:NFS:TLIMit (?)	2-105
SYSTem:COMMunicate:LAN:PING? (Query Only)	2-105
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:ADDRess (?)	2-106
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:FSYSTem (?)	2-107
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:NAME (?)	2-107
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:PROToCol (?)	2-108
SYSTem:COMMunicate:LAN:RDEVice[1 2 3][:STATe] (?)	2-109
SYSTem:COMMunicate:LAN[:SELf]:ADDRess (?)	2-109
SYSTem:COMMunicate:LAN[:SELf]:MADDRess? (Query Only)	
AWG400 series Only	2-110
SYSTem:COMMunicate:LAN[:SELf]:SMASk (?)	2-111
SYSTem:DATE (?)	2-112
SYSTem:ERRor? (Query Only)	
AWG500/600 series Only	2-113
SYSTem:ERRor[:NEXT]? (Query Only)	
AWG400 series Only	2-114
SYSTem:KDIRrection (?)	2-115
SYSTem:KEYBoard[:TYPE] (?)	2-115
SYSTem:KLOCK (?)	2-116
SYSTem:SECurity:IMMediate (No Query Form)	2-117
SYSTem:TIME (?)	2-117
SYSTem:UPTime? (Query Only)	2-118
SYSTem:VERSion? (Query Only)	2-119

*TRG (No Query Form) .....	2-119
TRIGger[:SEquence][:IMMediate] (No Query Form) .....	2-120
TRIGger[:SEquence]:IMPedance (?) .....	2-120
TRIGger[:SEquence]:LEVel (?) .....	2-121
TRIGger[:SEquence]:POLarity (?) .....	2-121
TRIGger[:SEquence]:SLOPe (?) .....	2-122
TRIGger[:SEquence]:SOURce (?) .....	2-123
TRIGger[:SEquence]:TIMer (?) .....	2-123
*TST? (Query Only) .....	2-124
*WAI (No Query Form) .....	2-125
<b>Retrieving Response Messages .....</b>	<b>2-127</b>
<b>Data Transfer .....</b>	<b>2-129</b>
Data File .....	2-129
About Waveform and Pattern Files .....	2-130
Data Transfer Procedures .....	2-138

## Status and Events

<b>Status and Event Reporting .....</b>	<b>3-1</b>
Status Reporting Structure .....	3-1
Registers .....	3-4
Status Registers .....	3-4
Enable Registers .....	3-8
Queues .....	3-10
Status and Event Processing Sequence .....	3-11
I/O Status and Event Screen .....	3-13
Synchronizing Execution .....	3-14
Messages .....	3-14
<b>Messages and Codes .....</b>	<b>3-15</b>
Command Errors .....	3-16
Execution Errors .....	3-18
Device Specific Errors .....	3-20
Query Errors .....	3-21
Power-On Events .....	3-21
User Request Events .....	3-21
Request Control Events .....	3-22
Operation Complete Events .....	3-22
Device Errors .....	3-23

## Examples

<b>Programming Examples .....</b>	<b>4-1</b>
-----------------------------------	------------

## Appendices

<b>Appendix A: Character Charts .....</b>	<b>A-1</b>
<b>Appendix B: GPIB Interface Specification .....</b>	<b>B-1</b>
Interface Functions .....	B-1
Interface Messages .....	B-3
<b>Appendix C: Network Interface Specification .....</b>	<b>C-1</b>
<b>Appendix D: SCPI Conformance Information .....</b>	<b>D-1</b>



**Appendix E: Factory Initialization Settings** ..... **E-1**

**Glossary and Index**

## List of Tables

<b>Table 2–1: BNF symbols and meanings</b> .....	<b>2–1</b>
<b>Table 2–2: Query response examples</b> .....	<b>2–3</b>
<b>Table 2–3: Parameter types used in syntax descriptions</b> .....	<b>2–4</b>
<b>Table 2–4: Functional groups in the AWG command set</b> .....	<b>2–17</b>
<b>Table 2–5: AWG Control commands</b> .....	<b>2–20</b>
<b>Table 2–6: Calibration commands</b> .....	<b>2–21</b>
<b>Table 2–7: Diagnostic commands</b> .....	<b>2–21</b>
<b>Table 2–8: Display commands</b> .....	<b>2–21</b>
<b>Table 2–9: Hardcopy commands</b> .....	<b>2–22</b>
<b>Table 2–10: Mass storage in AWG400/500/600 series</b> .....	<b>2–23</b>
<b>Table 2–11: Mass Memory commands</b> .....	<b>2–23</b>
<b>Table 2–12: Output channel</b> .....	<b>2–24</b>
<b>Table 2–13: Output commands</b> .....	<b>2–24</b>
<b>Table 2–14: Available source and markers</b> .....	<b>2–25</b>
<b>Table 2–15: Source commands</b> .....	<b>2–26</b>
<b>Table 2–16: Status commands</b> .....	<b>2–27</b>
<b>Table 2–17: Synchronization commands</b> .....	<b>2–27</b>
<b>Table 2–18: System commands</b> .....	<b>2–28</b>
<b>Table 2–19: Trigger commands</b> .....	<b>2–30</b>
<b>Table 2–20: Argument and available AWG</b> .....	<b>2–32</b>
<b>Table 2–21: Selecting run modes</b> .....	<b>2–50</b>
<b>Table 2–22: Self-test routines</b> .....	<b>2–58</b>
<b>Table 3–1: SBR bit functions</b> .....	<b>3–5</b>
<b>Table 3–2: SESR bit functions</b> .....	<b>3–6</b>
<b>Table 3–3: OCR bit functions</b> .....	<b>3–7</b>
<b>Table 3–4: QCR bit functions</b> .....	<b>3–7</b>
<b>Table 3–5: Definition of event codes</b> .....	<b>3–15</b>
<b>Table 3–6: Command errors</b> .....	<b>3–16</b>
<b>Table 3–7: Execution errors</b> .....	<b>3–18</b>
<b>Table 3–8: Device specific errors</b> .....	<b>3–20</b>
<b>Table 3–9: Query errors</b> .....	<b>3–21</b>
<b>Table 3–10: Power-on events</b> .....	<b>3–21</b>
<b>Table 3–11: User request events</b> .....	<b>3–21</b>
<b>Table 3–12: Request control events</b> .....	<b>3–22</b>

---

<b>Table 3–13: Operation complete events</b> .....	<b>3–22</b>
<b>Table 3–14: Device errors (AWG400 series)</b> .....	<b>3–23</b>
<b>Table 3–15: Device errors (AWG500/600 series)</b> .....	<b>3–30</b>
<b>Table A–1: The AWG character set</b> .....	<b>A–1</b>
<b>Table A–2: ASCII &amp; GPIB code chart</b> .....	<b>A–2</b>
<b>Table B–1: GPIB interface function implementation</b> .....	<b>B–1</b>
<b>Table B–2: AWG standard interface message</b> .....	<b>B–3</b>
<b>Table D–1: SCPI conformance information(AWG400 series)</b> .....	<b>D–1</b>
<b>Table D–2: SCPI conformance information(AWG500/600 series)</b> ...	<b>D–4</b>
<b>Table E–1: Factory initialization settings (AWG400 series)</b> .....	<b>E–1</b>
<b>Table E–2: Factory initialization settings (AWG500/600 series)</b> ....	<b>E–4</b>

## List of Figures

Figure 1–1: Common message elements .....	1–1
Figure 1–2: Functional groupings and alphabetical list of commands .....	1–2
Figure 1–3: Basic operation of status and events reporting .....	1–3
Figure 1–4: The floppy disk .....	1–3
Figure 1–5: GPIB connector location .....	1–4
Figure 1–6: How to stack GPIB connectors .....	1–5
Figure 1–7: Typical GPIB network configurations .....	1–6
Figure 1–8: Selecting the GPIB configuration and address .....	1–7
Figure 1–9: Ethernet port location (AWG400 series) .....	1–8
Figure 1–10: Ethernet port location (AWG500/600 series) .....	1–8
Figure 1–11: Setting the Network parameters .....	1–9
Figure 1–12: Message box to indicate the establishment of communication .. 1–10	
Figure 2–1: Example of SCPI subsystem hierarchy tree .....	2–2
Figure 2–2: Example of abbreviating a command .....	2–5
Figure 2–3: Example of chaining commands and queries .....	2–6
Figure 2–4: Example of omitting root and lower-level nodes in a chained message .....	2–6
Figure 2–5: Typical syntax diagrams .....	2–15
Figure 2–6: ABSTouch arguments and associated controls .....	2–35
Figure 2–7: ABSTouch arguments and Front panel key area .....	2–36
Figure 2–8: ABSTouch arguments and Trigger, Output controls ....	2–36
Figure 2–9: Retrieving response messages .....	2–127
Figure 2–10: The Waveform file format .....	2–131
Figure 2–11: The Pattern File format .....	2–132
Figure 2–12: The Sequence File format .....	2–134
Figure 2–13: The Equation File format .....	2–136
Figure 2–14: The Code Convert File format .....	2–137
Figure 3–1: Error and Event handling process overview .....	3–2
Figure 3–2: The Status Byte Register (SBR) .....	3–5
Figure 3–3: The Standard Event Status Register (SESR) .....	3–6
Figure 3–4: The Operation Condition Register (OCR) .....	3–7
Figure 3–5: The Questionable Condition Register (QCR) .....	3–7
Figure 3–6: The Event Status Enable Register (ESER) .....	3–8

---

<b>Figure 3–7: The Service Request Enable Register (SRER) . . . . .</b>	<b>3–9</b>
<b>Figure 3–8: The Operation Enable Register (OENR) . . . . .</b>	<b>3–9</b>
<b>Figure 3–9: The Questionable Enable Register (QENR) . . . . .</b>	<b>3–9</b>
<b>Figure 3–10: Status and Event processing sequence — Operation status block . . . . .</b>	<b>3–11</b>
<b>Figure 3–11: Status and Event processing sequence — Questionable status block . . . . .</b>	<b>3–11</b>
<b>Figure 3–12: Status and Event processing sequence — Standard/Event status block . . . . .</b>	<b>3–12</b>
<b>Figure 3–13: Status and Event screen . . . . .</b>	<b>3–13</b>
<b>Figure 4–1: Equipment needed to run the GPIB example programs .</b>	<b>4–1</b>



# Preface

This is the programmer manual for the AWG400/500/600 Series Arbitrary Waveform Generators. This manual provides information necessary for operating the instrument over both the General Purpose Interface Bus (GPIB) and Ethernet interfaces.

This manual provides the following information:

- The *Getting Started* section describes how to connect and set up the waveform generator for remote operation.
- The *Syntax and Commands* section defines the command syntax and processing conventions and describes each command in the waveform generator command set.
- The *Status and Events* section explains the status information and event messages reported by the waveform generator.
- The *Programming Examples* section describes how to use the Sample Program floppy disk supplied with the waveform generator.
- The *Appendices* section contains various tables of reference information.
- The *Glossary and Index* section contains a glossary of common terms and an index to this manual.

## Related Manuals

Other documentation for the waveform generator includes:

- The *AWG410, 420 & AWG430 200MHz Arbitrary Waveform Generator User Manual* (Tektronix part number 070-A809-50) describes the operation of the instrument.
- The *AWG510 & AWG520 Arbitrary Waveform Generator User Manual* (Tektronix part number 071-0099-50) describes the operation of the instrument.
- The *AWG610 Arbitrary Waveform Generator User Manual* (Tektronix part number 071-0554-50) describes the operation of the instrument.
- The *AWG510 & AWG520 Arbitrary Waveform Generator Service Manual* (Tektronix part number 071-0101-50) provides information for maintaining and servicing the waveform generator.

- The *AWG610 Arbitrary Waveform Generator Service Manual* (Tektronix part number 071-0556-50) provides information for maintaining and servicing the waveform generator.





# Getting Started



# Getting Started

The AWG400/500/600 Series Arbitrary Waveform Generator has GPIB and 10Base-T/100Base-TX (AWG400 series only) Ethernet interface capability. You can write computer programs that remotely set the front panel controls or that transfer waveform data.

To help you get started with programming the waveform generator, this section includes the following subsections:

- *Manual Overview* – summarizes the type of programming information contained in each major section in this manual.
- *Setting Up Remote Communications Using GPIB* – describes how to connect the waveform generator to a controller through the GPIB interface, and how to set the appropriate front panel controls.
- *Setting Up Remote Communications Using Ethernet* – describes how to connect the waveform generator to a controller using the Ethernet interface and how to set the appropriate front panel controls.

## Manual Overview

A summary of the information provided in each major section of this manual follows:

### Syntax and Commands

The *Command Syntax* subsection, which begins on page 2–1, describes the structure and content of the messages your program sends to the waveform generator. You can use the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common Commands. Figure 1–1 is an example of the syntax and command parts diagrams used in the *Command Syntax* subsection.

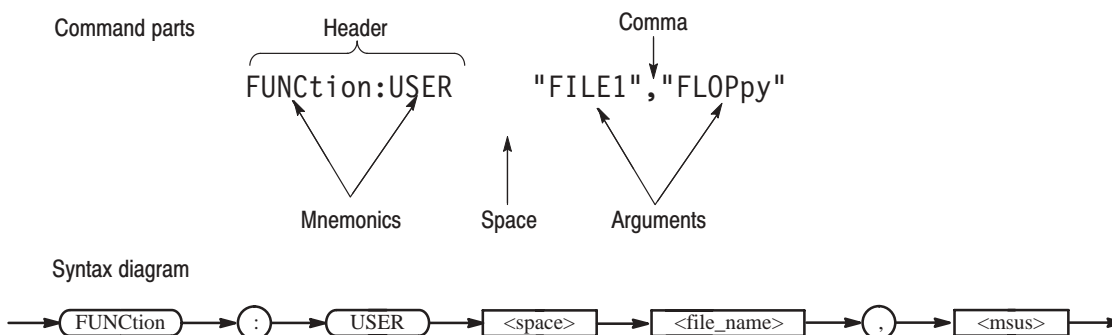
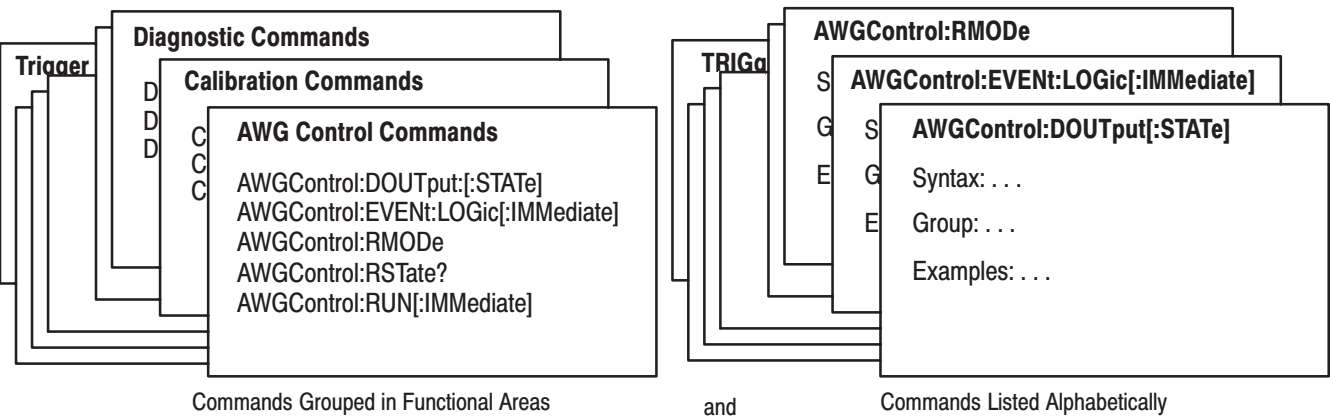


Figure 1–1: Common message elements

The *Command Syntax* subsection also describes the result of each command, and provides examples of how you might use it. The *Command Groups* subsection, which begins on page 2–17, provides a command list by functional area. The *Command Descriptions* subsection, which begins on page 2–31, arranges commands alphabetically. Figure 1–2 illustrates the two kinds of command lists.

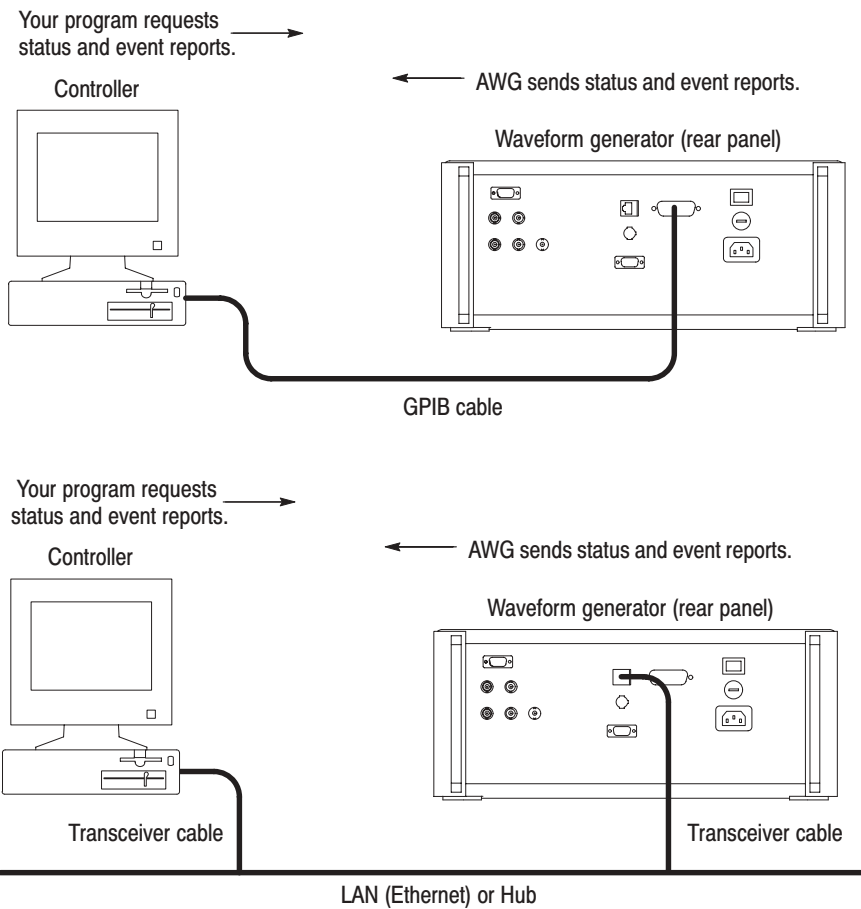


**Figure 1–2: Functional groupings and alphabetical list of commands**

**Status and Events Reporting**

The program may request information from the waveform generator. The waveform generator provides information in the form of status and error messages. Figure 1–3 on page 1–3 illustrates the basic operation of this system.

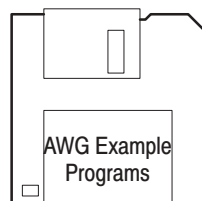
The *Status and Events Reporting* subsection, which begins on page 3–1, describes how to use the status reporting functions that conform to SCPI and IEEE–488.2 in your programs.



**Figure 1-3: Basic operation of status and events reporting**

## Programming Examples

The *Programming Examples* section, which begins on page 4-1, provides some sample waveform generator programs. A floppy disk (see Figure 1-4) is supplied with this manual. The disk contains a Microsoft Visual C++ and Visual BASIC source-code version of each program.



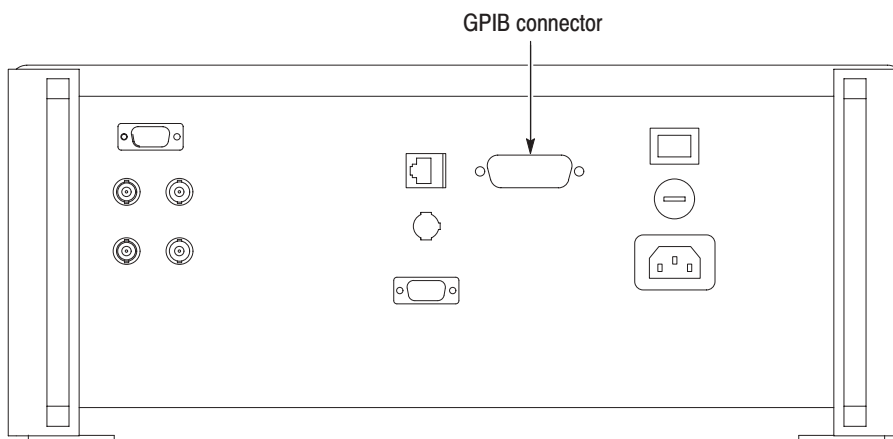
**Figure 1-4: The floppy disk**

## Setting Up Remote Communications Using GPIB

For remote operations, the instrument must be connected to the controller.

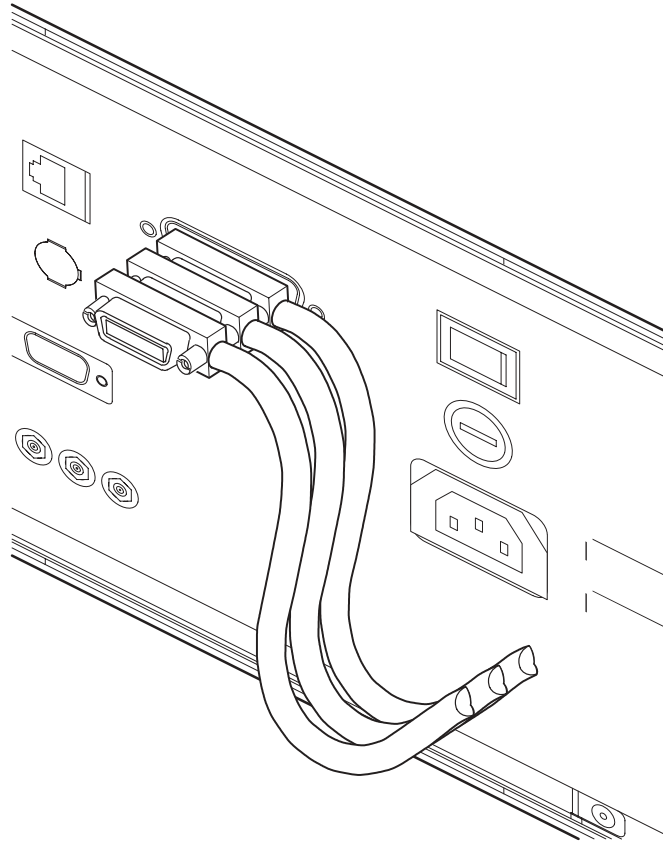
The waveform generator has a 24-pin GPIB connector on its rear panel, as shown in Figure 1-5. This connector has a D-type shell and conforms to IEEE Std 488.1-1987.

Attach an IEEE Std 488.1-1987 GPIB cable (Tektronix Part Number 012-0991-xx) to the GPIB connector.



**Figure 1-5: GPIB connector location**

Stack GPIB connectors, if needed, as shown in Figure 1-6.

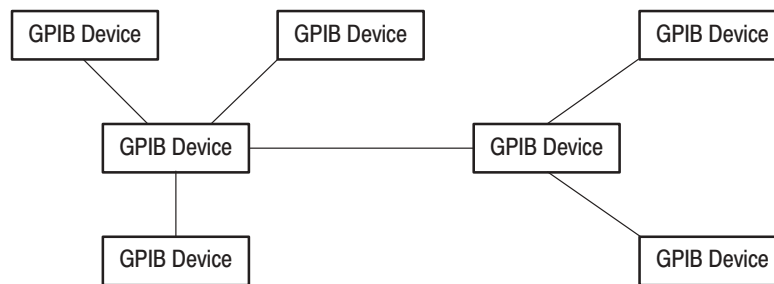


**Figure 1-6: How to stack GPIB connectors**

**GPIB Requirements**

Follow these rules when you use your waveform generator with a GPIB network:

- Assign a unique device address to each device on the bus. Two devices can not share the same device address.
- Do not connect more than 15 devices to one bus.
- Connect one device for every 2 meters (6 feet) of cable used.
- Do not use more than 20 meters (65 feet) of cable to connect devices to a bus.
- While using the network, turn on at least two-thirds of the devices on the network.
- Connect the devices on the network in a star or linear configuration, as shown in Figure 1–7. Do not use loop or parallel configurations.



**Figure 1–7: Typical GPIB network configurations**

---

**NOTE.** Appendix C: Network Interface Specification *provides more information about the GPIB configuration of the waveform generator.*

---

**Setting the GPIB Parameters**

You must set the GPIB parameters of the waveform generator to match the configuration of the bus. Follow the steps below to set up the waveform generator for the GPIB interface.

1. Press the **UTILITY** button to display the Utility screen.
2. Press the **Comm** menu button at the bottom of the screen.
3. Move the cursor to the **Remote Control** field using the up/down ( $\uparrow/\downarrow$ ) arrow buttons, then select **GPIB** using the left/right ( $\leftarrow/\rightarrow$ ) arrow buttons.
4. Move the cursor to the **GPIB Configuration** field using the up/down ( $\uparrow/\downarrow$ ) arrow buttons, then select **Talk/Listen** using either the general purpose knob or the left/right ( $\leftarrow/\rightarrow$ ) arrow buttons. See Figure 1–8 on page 1–7.



5. Move the cursor to the **GPIB Address** field using the down (↓) arrow button. Set the address using either the general purpose knob or the keypad.

Clock: 100.0000MS/s		Run Mode: Continuous		Stopped									
Remote Control:					GPIB Network <input checked="" type="radio"/>								
<u>GPIB</u>		Configuration:		Talk/Listen Controller Off Bus									
Address:		1											
<u>Network</u>		DHCP Client:		Disabled Enabled									
IP Address:		192.168.0.3											
Subnet Mask:		255.255.240.0											
MAC Address:		xx:00:11:22:33:44											
		Destination Network		Gateway Address									
Gateway 1:		<input type="text"/>		<input type="text"/>									
Gateway 2:		<input type="text"/>		<input type="text"/>									
Gateway 3:		<input type="text"/>		<input type="text"/>									
FTP Server:		Disabled Enabled											
					Comm								
					Execute Ping...								
					Edit...								
System		Disk		Comm		Network		Status		Diag		Service	

**Figure 1-8: Selecting the GPIB configuration and address**

The waveform generator is set up for bidirectional communication with your controller. Do the following to isolate the waveform generator from the bus:

Select **Off Bus** in the **GPIB Configuration** field.

This selection disables all communication with the controller.

## Setting Up Remote Communications Using Ethernet

**NOTE.** For remote operations, the instrument must be connected to the controller.

The waveform generator has an Ethernet (10Base-T/100Base-Tx) port on the rear panel (100Base-Tx is AWG400 series only) as shown in Figures 1–9 and 1–10.

Attach an Ethernet cable to the Ethernet port.

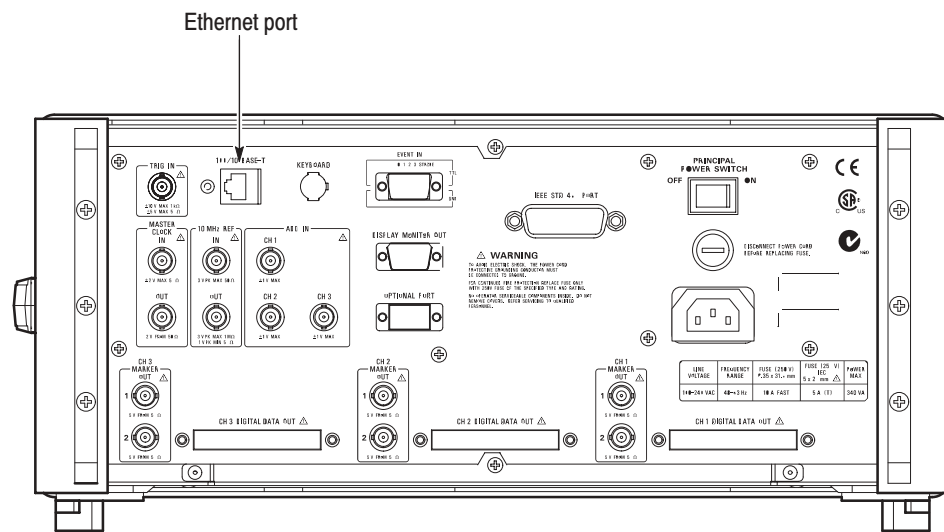


Figure 1–9: Ethernet port location (AWG400 series)

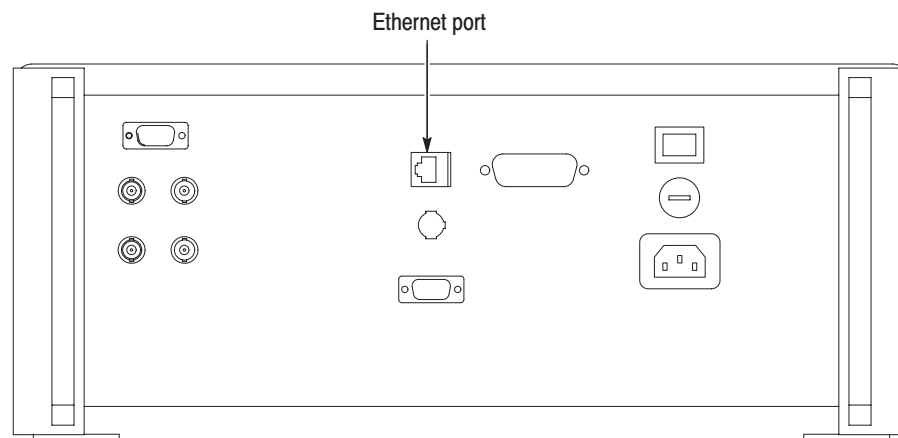


Figure 1–10: Ethernet port location (AWG500/600 series)

## Setting the Network Parameters

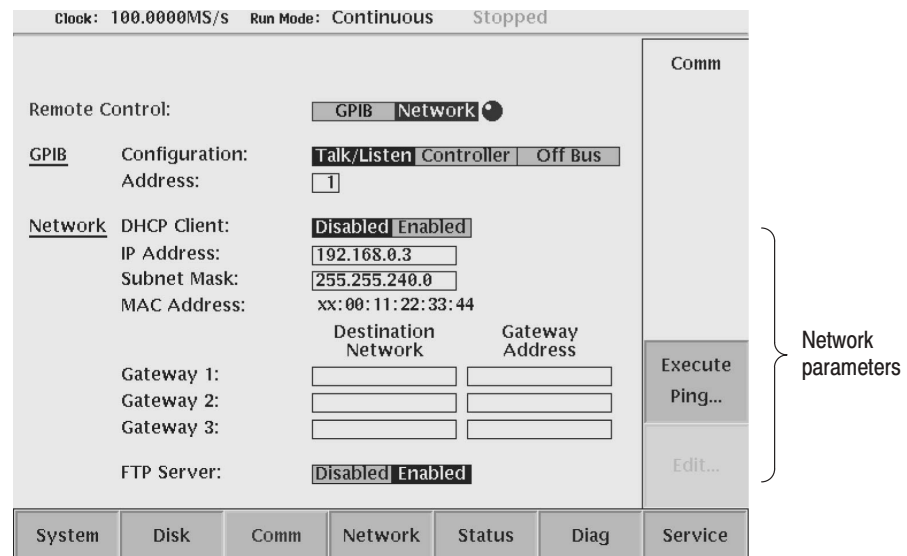
You must set the network parameters of the waveform generator to match the configuration of the network. After you have set these parameters, you can control the waveform generator through the Ethernet interface.

1. Press the **UTILITY** button to display the Utility screen.
2. Press the **Comm** menu button at the bottom of the screen.
3. Move the cursor to the **Remote Control** field using the up/down ( $\uparrow/\downarrow$ ) arrow buttons, then select **Network** using the left/right ( $\leftarrow/\rightarrow$ ) arrow buttons.
4. Move the cursor to the **Network IP Address** field using the up/down ( $\uparrow/\downarrow$ ) arrow buttons, then set the address using the keypad. See Figure 1–11 on page 1–9.
5. If necessary, use the **Subnet Mask** field to set the address.
6. If necessary, use the **Destination Network** and **Gateway Address** fields to set the destination network and the address.

You need to set the gateway address when the remote computers are connecting to another network that is connected to the network via gateway. You can set up to three gateways.

Setting the FTP server to **Enabled** for access to the hard disk system of the instrument from a remote computer.

If you are not familiar with the network setup, consult with your network administrator.



**Figure 1–11: Setting the Network parameters**

### Testing the Network Connection

After completing the connection and settings, verify that the waveform generator can recognize the network and the remote computers, or if the network can recognize the waveform generator. Following these steps to use the ping command to verify that the instrument can communicate with the network:

1. Press the **UTILITY** button to display the Utility screen.
2. Press the **Network** or **Comm** bottom menu button.
3. Press the **Execute Ping** side button to display a dialog box.
4. Enter the IP address of the remote computer in the dialog box, and then push the **OK** side button.

The ping command sends a packet to the remote computer specified by the IP address. When the computer receives the packet, it sends the packet back to the sender (waveform generator).

When the waveform generator can communicate with the remote computer through the network the message in Figure 1–12 displays. If communication failed, the message box displays an error message such as "no response from...".

5. Repeat steps 2 and 3 to verify the connection for other remote computers on the network.



**Figure 1–12: Message box to indicate the establishment of communication**



# Syntax and Commands



# Command Syntax

This section contains general information about command structure and syntax usage. You should familiarize yourself with this material before using the waveform generator command descriptions.

This manual describes commands and queries using Backus-Naur Form (BNF) notation. Table 2-1 defines standard BNF symbols.

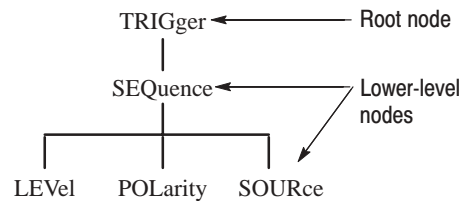
**Table 2-1: BNF symbols and meanings**

Symbol	Meaning
< >	Defined element
::=	Is defined as
	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated
( )	Comment

## SCPI Commands and Queries

The waveform generator uses a command language based on the SCPI standard. The SCPI (Standard Commands for Programmable Instruments) standard was created by a consortium to provide guidelines for remote programming of instruments. These guidelines provide a consistent programming environment for instrument control and data transfer. This environment uses defined programming messages, instrument responses and data formats that operate across all SCPI instruments, regardless of manufacturer.

The SCPI language is based on a hierarchical or tree structure that represents a subsystem (see Figure 2–1). The top level of the tree is the root node; it is followed by one or more lower-level nodes.



**Figure 2-1: Example of SCPI subsystem hierarchy tree**

You can create commands and queries from these subsystem hierarchy trees. Commands specify actions for the instrument to perform. Queries return measurement data and information about parameter settings.



**Creating Commands**

SCPI commands are created by stringing together the nodes of a subsystem hierarchy and separating each node by a colon.

In Figure 2–1 on page 2–2, TRIGger is the root node and SEQuence, LEVel, POLarity, and SOURce are lower-level nodes. To create an SCPI command, start with the root node TRIGger and move down the tree structure adding nodes until you reach the end of a branch. Most commands and some queries have parameters; you must include a value for these parameters. The command descriptions, which begin on page 2–31, list the valid values for all parameters.

For example, TRIGger:SEQuence:SOURce EXTernal is a valid SCPI command created from the hierarchy tree in Figure 2–1 on page 2–2.

**Creating Queries**

To create a query, start at the root node of a tree structure, move down to the end of a branch, and add a question mark. TRIGger:SEQuence:SOURce? is an example of a valid SCPI query using the hierarchy tree in Figure 2–1 on page 2–2.

**Query Responses**

The query causes the waveform generator to return information about its status or settings. When a query is sent to the waveform generator, only the values are returned. When the returned value is a mnemonic, it is noted in abbreviated format, as shown in Table 2–2.

**Table 2–2: Query response examples**

Query	Response
SOURce:VOLTage:AMPLitude?	1.000
AWGControl:RMODE?	CONT

A few queries also initiate an operation action before returning information. For example, the \*CAL? query runs a calibration.

**Parameter Types**

Parameters are indicated by angle brackets, such as <file\_name>. There are several different types of parameters, as listed in Table 2–3. The parameter type is listed after the parameter. Some parameter types are defined specifically for the AWG400/500/600 series command set and some are defined by SCPI.

**Table 2–3: Parameter types used in syntax descriptions**

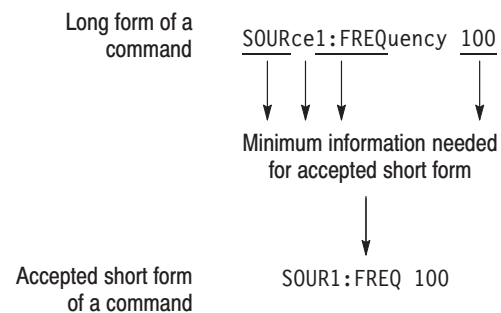
Parameter Type	Description	Example
arbitrary block	A block of data bytes	#512234xxxxx... where 5 indicates that the following 5 digits (12234) specify the length of the data in bytes; xxxxx... indicates the data  or #0xxxxx...<LF><&EOI>
boolean	Boolean numbers or values	ON or $\neq$ 0 OFF or 0
discrete	A list of specific values	MIN, MAX
binary	Binary numbers	#B0110
octal	Octal numbers	#Q75, #Q3
hexadecimal	Hexadecimal numbers (0–9, A– F)	#HAA, #H1
NR1 numeric	Integers	0, 1, 15, -1
NR2 numeric	Decimal numbers	1.2, 3.141516, -6.5
NR3 numeric	Floating point numbers	3.1415E-9, -16.1E5
NRf numeric	Flexible decimal number that may be type NR1, NR2, or NR3	See NR1, NR2, NR3 examples in this table
string	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

**Special Characters**

The Line Feed (LF) character or the New Line (NL) character (ASCII 10), and all characters in the range of ASCII 127-255 are defined as special characters. These characters are used in arbitrary block arguments only; using these characters in other parts of any command yields unpredictable results.

**Abbreviating Commands, Queries, and Parameters**

You can abbreviate most SCPI commands, queries, and parameters to an accepted short form. This manual shows these commands as a combination of upper and lower case letters. The upper case letters indicate the accepted short form of a command, as shown in Figure 2–2. The accepted short form and the long form are equivalent and request the same action of the instrument.



**Figure 2–2: Example of abbreviating a command**

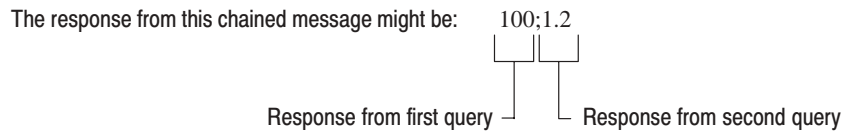
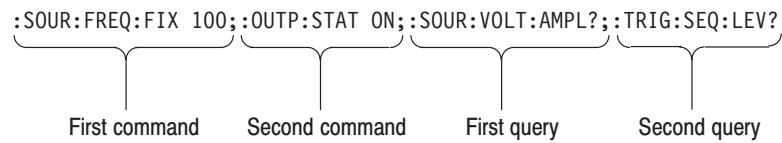
---

**NOTE.** The numeric suffix of a command or query may be included in either the long form or short form; the AWG400/500/600 series will default to “1” if no suffix is used.

---

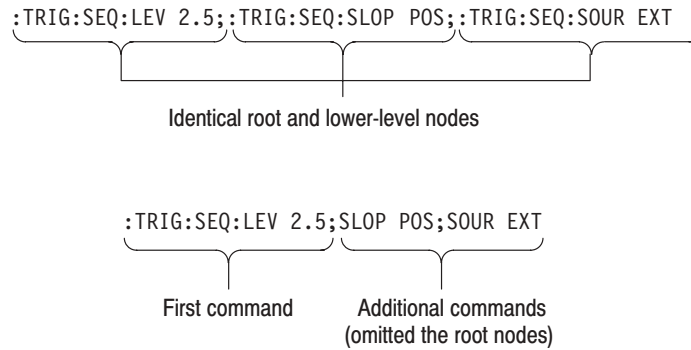
### Chaining Commands and Queries

You can chain several commands or queries together into a single message. To create a chained message, first create a command or query, then add a semicolon (;), and finally add more commands or queries and semicolons until you are done. If the command following a semicolon is a root node, precede it with a colon (:). Figure 2–3 illustrates a chained message consisting of several commands and queries. The chained message should end in a command or query, not a semicolon. Responses to any queries in your message are separated by semicolons.



**Figure 2-3: Example of chaining commands and queries**

If a command or query has the same root and lower-level nodes as the previous command or query, you can omit these nodes. In Figure 2–4, the second command has the same root node (SEQUence) as the first command, so these nodes can be omitted.



**Figure 2-4: Example of omitting root and lower-level nodes in a chained message**

**Unit and SI Prefix**

If the decimal numeric argument refers to voltage, frequency, impedance, or time, you can express it using SI units instead of using the scaled explicit point input value format <NR3>. (SI units are units that conform to the System International d'Unites standard.) For example, you can use the input format 200 mV or 1.0 MHz instead of 200.0E-3 or 1.0E+6, respectively, to specify voltage or frequency.

You can omit the unit, but you must include the SI unit prefix. You can use either upper or lowercase units.

V or v for voltage

Hz, HZ, or hz for frequency

ohm, OHM, or Ohm for impedance

s or S for time

dbm, DBM, or Dbm for power ratio (AWG400/500 series only)

In the case of angle, you can use RADian and DEGree. The default unit is RADian.

The SI prefixes, which must be included, are shown below. Note that either lower or upper case prefixes can be used.

SI prefix *	p/P	n/N	u/U	m/M	k/K	M/M	g/G
Corresponding power	10 <sup>-12</sup>	10 <sup>-9</sup>	10 <sup>-6</sup>	10 <sup>-3</sup>	10 <sup>3</sup>	10 <sup>6</sup>	10 <sup>9</sup>

\* Note that the prefix m/M indicates 10<sup>-3</sup> when the decimal numeric argument denotes voltage or time, but indicates 10<sup>6</sup> when it denotes frequency.

\* Note that the prefix u/U is used instead of "μ".

Use **mV** for **V**, and **MHz** for **Hz**.

**General Rules**

Here are three general rules for using SCPI commands, queries, and parameters:

- You can use single ( ' ') or double ( " ") quotation marks for quoted strings, but you cannot use both types of quotation marks for the same string.

correct:            "This string uses quotation marks correctly."

correct:            'This string also uses quotation marks correctly.'

incorrect:          "This string does not use quotation marks correctly.'

- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.

                      :OUTPUT:FILTER:LPASS:FREQUENCY 200MHZ

is the same as

                      output:filter:lpass:frequency 200MHz

and

                      OUTPUT:filter:LPASS:frequency 200MHz

---

**NOTE.** *Literal strings (quoted) are case sensitive. For example: file names.*

---

- No embedded spaces are allowed between or within nodes.

correct:            OUTPUT:FILTER:LPASS:FREQUENCY 200MHZ

incorrect:          OUTPUT: FILTER: LPASS:FREQUENCY 200MHZ

## IEEE 488.2 Common Commands

ANSI/IEEE Standard 488.2 defines the codes, formats, protocols, and usage of common commands and queries used on the interface between the controller and the instruments. The waveform generator complies with this standard.

The syntax for an IEEE 488.2 common command is an asterisk (\*) followed by a command and, optionally, a space and parameter value. The syntax for an IEEE 488.2 common query is an asterisk (\*) followed by a query and a question mark. All of the common commands and queries are included in the *Syntax and Commands* section of this manual. The following are examples of common commands:

- \*ESE 16
- \*CLS

The following are examples of common queries:

- \*ESR?
- \*IDN?

## Constructed Mnemonics

Some command headers list a range of mnemonics. When constructing the command, you select one mnemonic from the list. You then use the mnemonic in the command just as you do any other mnemonic. Mnemonic ranges can be presented in any of the following formats:

**MNEMonic**[a|b|c]. The values a, b, and c represent the actual list of valid selections. You cannot list more than one value.

For example, for the command **SYSTem:COMMUnicate:LAN:GATe-way**[1|2|3]:ADDRes, the gateway mnemonic could be any of the following: **GATe**way1, **GATe**way2, or **GATe**way3. Therefore, a valid usage of this command would be: **SYSTem:COMMUnicate:LAN:GATe**way1:ADDRes.

**MNEMonic**<n>. The value of <n> is the upper range of valid suffixes. If the numeric suffix is omitted, the waveform generator uses the default value of “1”.

### Source Channel Mnemonics

These commands specify the source channel to use as a mnemonic in the header.

Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
SOURce1	CH1 signal of waveform generator	○	○	○	○	○	○
SOURce2	CH2 signal of waveform generator	×	○	○	×	○	×
SOURce3	CH3 signal of waveform generator	×	×	○	×	×	×
SOURce4	CH1 digital data	Op03 only	Op03 only	Op03 only	×	×	×
SOURce5	CH2 digital data	×	Op03 only	Op03 only	Op03 only	Op03 only	×
SOURce6	CH3 digital data	×	×	Op03 only	×	×	×
SOURce7	CH1 noise ADD signal	○	○	○			
	Noise ADD signal				○	○	×
SOURce8	CH1 Ext. ADD signal	○	○	○	○	○	×
SOURce9	CH2 noise ADD signal	×	○	○	×	×	×
SOURce10	CH2 Ext. ADD signal	×	○	○	×	×	×



Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
SOURce11	CH3 noise ADD signal	X	X	O	X	X	X
SOURce12	CH3 Ext. ADD signal	X	X	O	X	X	X

O:usable  
X:not used  
OpXX:usable OptXX only

### Output Channel Mnemonics

These commands specify the output channel to use as a mnemonic in the header.

Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
OUTPut1	CH1 analog signal output	O	O	O	O	O	O
OUTPut2	CH2 analog signal output	X	O	O	X	O	X
OUTPut3	CH3 analog signal output	X	X	O	X	X	X
OUTPut4	CH1 digital data output	Op03 only	Op03 only	Op03 only	X	X	X
OUTPut5	CH2 digital data output	X	Op03 only	Op03 only			
	digital data output				Op03 only	Op03 only	X
OUTPut6	CH3 digital data output	X	X	Op03 only	X	X	X
OUTPut7	Noise ADD signal	X	X	X	O	O	X

O:usable  
X:not used  
OpXX:usable OptXX only

### Direct D/A Output Mnemonics

These commands specify the direct D/A converter output to use as a mnemonic in the header.

Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
DOUtput1	Direct output from CH1 D/A converter	○	○	○	○	○	○
DOUtput2	Direct output from CH2 D/A converter	X	○	○	X	○	X
DOUtput3	Direct output from CH3 D/A converter	X	X	○	X	X	X

○:usable  
X:not used

### Gateway Mnemonics

These commands specify the gateway to use as a mnemonic in the header.

Symbol	Meaning
GATeway1	Gateway 1
GATeway2	Gateway 2
GATeway3	Gateway 3

### Marker Mnemonics

These commands specify the marker to use as a mnemonic in the header.

Symbol	Meaning
MARKer1	The signal for the marker 1 (AWG500/600 series only)
MARKer2	The signal for the marker 2 (AWG500/600 series only)

### Remote Device Mnemonics

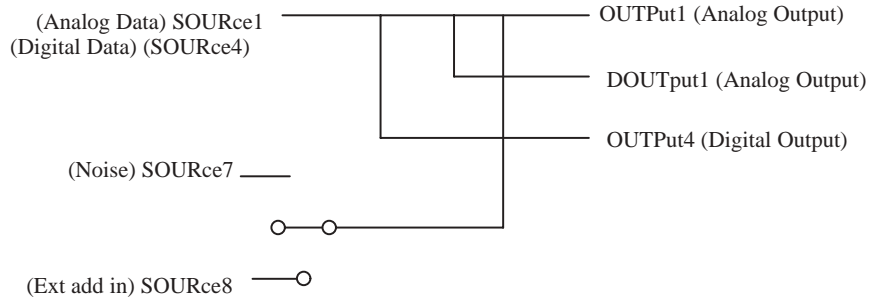
These commands specify the remote device to use as a mnemonic in the header.

Symbol	Meaning
RDEVice1	Network drive 1
RDEVice2	Network drive 2
RDEVice3	Network drive 3

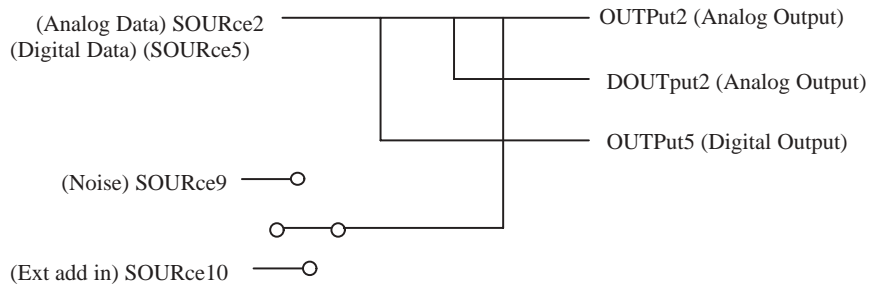
**Source to Output Connections**

The following illustrations shows the source to output connections for the AWG400 series, AWG510, AWG520, and AWG610 instruments.

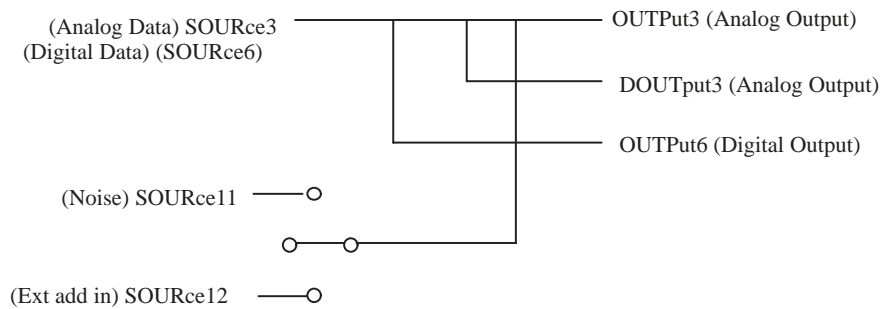
**AWG400 Series  
CH1**



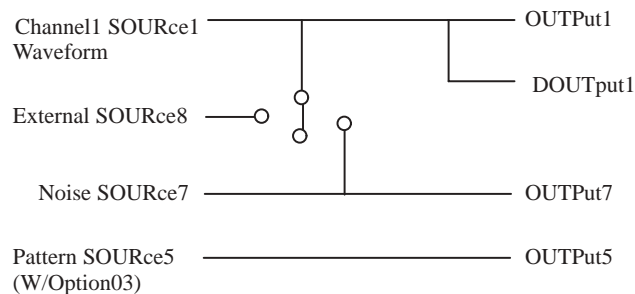
**CH2**



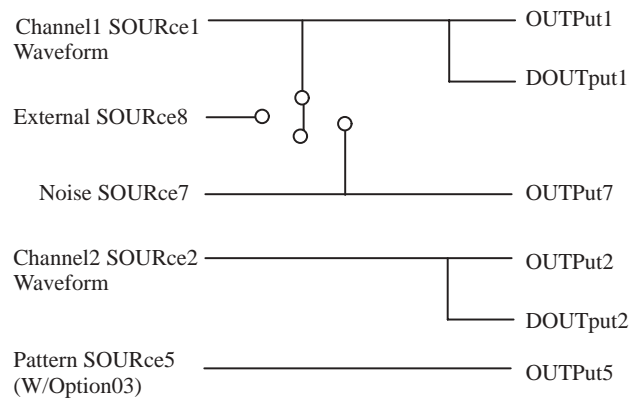
**CH3**



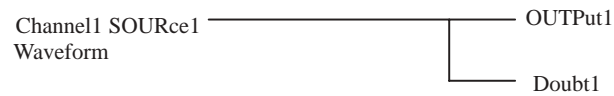
**AWG510**



### AWG520



### AWG610



## Syntax Diagrams

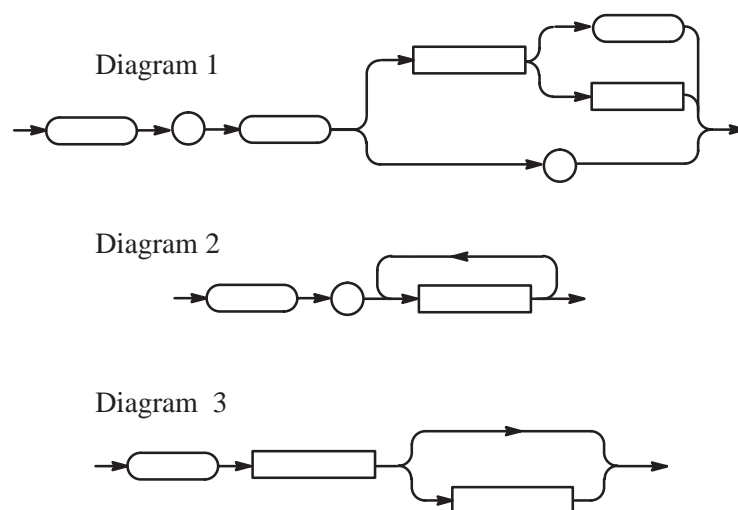
The syntax of each command and query is explained by both syntax diagrams and BNF notation. Figure 2–5 shows some typical syntax diagram structures. The syntax diagrams are described by the following symbols and notation:

- Oval symbols contain literal elements, such as a command or query header and a nonquoted string argument.
- Circle symbols contain separators or special symbols, such as (:), (,), and (?).
- Box symbols contain the defined element, such as <NR1>.
- Arrow symbols connect elements to show the paths that can be taken through the diagram and, thereby, the order in which the elements can be sent in a command structure.
- Parallel paths show that only one of the paths can be taken in the command. See diagram 1 in Figure 2–5.
- A loop around an element(s) shows the element can be repeated. See diagram 2 in Figure 2–5.
- A path around a group of elements shows that those elements are optional. See diagram 3 in Figure 2–5.

---

**NOTE.** The unit and SI prefix that can be added to decimal numeric arguments are not described in the syntax diagram. See Unit and SI Prefix on page 2–7.

---



**Figure 2–5: Typical syntax diagrams**



# Command Groups

This section lists commands in two ways, by functional groups and alphabetically. The functional group list starts below. The alphabetical list provides more detail on each command and starts on page 2–31.

The GPIB interface of AWG400 Series conforms to SCPI (Standard Commands for Programmable Instruments) 1999.0 and IEEE Std 488.2–1987, except where noted.

The GPIB interface of AWG500/600 Series conforms to SCPI (Standard Commands for Programmable Instruments) 1995.0 and IEEE Std 488.2–1987, except where noted.

## Functional Groups

Table 2–4 lists the functional groups into which the AWG400/500/600 Series Arbitrary Waveform Generator (AWG) commands are classified.

**Table 2–4: Functional groups in the AWG command set**

Group	Function
AWG Control	Control operating mode
Calibration	Perform calibration
Diagnostic	Control self-test routines
Display	Control the presentation of information on the front panel display
Hardcopy	Dump the whole display into the file on the mass storage
Mass Memory	Control file operations on the mass storage
Output	Control the characteristics of the waveform output port
Source	Set waveform and marker output parameters, such as frequency and level
Status	Set and query the registers and queues of the reporting system
Synchronization	Control operation complete and pending command execution
System	Control miscellaneous instrument functions such as LAN, security, and time
Trigger	Synchronize the waveform generator actions with events

## Command Quick Reference

The next page lists all the commands in each functional group and can be copied for use as a quick reference. The minimum accepted character string for each command is shown in uppercase characters.

### AWG Control commands

AWGControl:CLOCK:SOURce (?)  
 AWGControl:DOUTput<x>:STATe (?)  
 AWGControl:ENHanced:SEQUence:JMODE(?)  
 AWGControl:EVENT:LOGic:IMMEDIATE  
 AWGControl:EVENT:SOFTWARE:IMMEDIATE  
 AWGControl:EVENT:TABLE:IMMEDIATE  
 AWGControl:FG:FREQUENCY:CW|FIXed (?)  
 AWGControl:FG<n>:FUNCTION:SHAPE (?)  
 AWGControl:FG<n>:PHASE:ADJUST (?)  
 AWGControl:FG<n>:POLARITY (?)  
 AWGControl:FG<n>:PULSE:DCYCLE (?)  
 AWGControl:FG:STATE (?)  
 AWGControl:FG<n>:VOLTage:LEVEL:IMMEDIATE:AMPLITUDE (?)  
 AWGControl:FG<n>:VOLTage:LEVEL:IMMEDIATE:OFFSET (?)  
 AWGControl:RMODE (?)  
 AWGControl:RSTATE?  
 AWGControl:RUN:IMMEDIATE  
 AWGControl:SREStore  
 AWGControl:SSAVE  
 AWGControl:STOP:IMMEDIATE

### Calibration commands

\*CAL?  
 CALibration:ALL (?)

### Diagnostic commands

DIAGnostic:DATA?  
 DIAGnostic:IMMEDIATE (?)  
 DIAGnostic:SElect (?)  
 \*TST?

### Display commands

ABSTouch  
 DISPlay:BRIGhtness (?)  
 DISPlay:ENABle (?)  
 DISPlay:HIghLight:COLor (?)

### Hardcopy commands

HCOPY:DESTination  
 HCOPY:DEvice:COLor (?)  
 HCOPY:DEvice:LANGuage (?)  
 HCOPY:IMMEDIATE  
 HCOPY:SDUMp:IMMEDIATE

### Mass memory commands

MMEemory:CATalog?  
 MMEemory:CDIRECTory (?)  
 MMEemory:CLOSE  
 MMEemory:COPY

MMEemory:DATA (?)  
 MMEemory:DELeTe  
 MMEemory:FEED (?)  
 MMEemory:INITialize  
 MMEemory:MDIRECTory  
 MMEemory:MOVE  
 MMEemory:MSIS (?)  
 MMEemory:NAME (?)  
 MMEemory:OPEN

### Output commands

OUTPut<x>:FILTer:LPASs:FREQUENCY (?)  
 OUTPut<x>:ISTATE (?)  
 OUTPut<x>:STATe (?)

### Source commands

SOURce<x>:COMBine:FEED (?)  
 SOURce<x>:FREQUENCY:CW|FIXed (?)  
 SOURce<x>:FUNCTion:USER (?)  
 SOURce<x>:MARKer<y>:DELay (?)  
 SOURce<x>:MARKer<y>:VOLTage:LEVEL:IMMEDIATE:HIGH (?)  
 SOURce<x>:MARKer<y>:VOLTage:LEVEL:IMMEDIATE:LOW (?)  
 SOURce<x>:POWER:LEVEL:IMMEDIATE:AMPLITUDE (?)  
 SOURce<x>:ROSCillator:SOURce (?)  
 SOURce<x>:SKEW (?)  
 SOURce<x>:VOLTage:LEVEL:IMMEDIATE:AMPLITUDE (?)  
 SOURce5:VOLTage:LEVEL:IMMEDIATE:HIGH (?)  
 SOURce5:VOLTage:LEVEL:IMMEDIATE:LOW (?)  
 SOURce<x>:VOLTage:LEVEL:IMMEDIATE:OFFSET (?)

### Status commands

\*CLS  
 \*ESE (?)  
 \*ESR?  
 \*PSC (?)  
 \*SRE (?)  
 STATus:OPERation:CONDition?  
 STATus:OPERation:ENABle (?)  
 STATus:OPERation:EVENT?  
 STATus:PRESet  
 STATus:QUESTionable:CONDition?  
 STATus:QUESTionable:ENABle (?)  
 STATus:QUESTionable:EVENT?  
 \*STB?

### Synchronization commands

\*OPC (?)  
 \*WAI



**System commands**

\*IDN?  
 \*OPT?  
 \*RST  
 SYSTem:BEEPer:IMMediate  
 SYSTem:COMMunicate:LAN:DHCP:CLient:LEASe:TIME (?)  
 SYSTem:COMMunicate:LAN:DHCP:CLient:STATe (?)  
 SYSTem:COMMunicate:LAN:FTP:SERVer:STATe (?)  
 SYSTem:COMMunicate:LAN:FTP:SERVer:VERSion (?)  
 SYSTem:COMMunicate:LAN:GATeway<x>:ADDRess (?)  
 SYSTem:COMMunicate:LAN:NFS:TLMit (?)  
 SYSTem:COMMunicate:LAN:PING?  
 SYSTem:COMMunicate:LAN:RDEVice<x>:ADDRess (?)  
 SYSTem:COMMunicate:LAN:RDEVice<x>:FSYSem (?)  
 SYSTem:COMMunicate:LAN:RDEVice<x>:NAME (?)  
 SYSTem:COMMunicate:LAN:RDEVice<x>:PROTocol (?)  
 SYSTem:COMMunicate:LAN:RDEVice<x>:STATe (?)  
 SYSTem:COMMunicate:LAN:SELF:ADDRess (?)  
 SYSTem:COMMunicate:LAN:SELF:MADDRess (?)  
 SYSTem:COMMunicate:LAN:SELF:SMASK (?)

SYSTem:DATE (?)  
 SYSTem:ERRor:NEXT?  
 SYSTem:KDIRection (?)  
 SYSTem:KEYBoard:TYPE (?)  
 SYSTem:KLOCK (?)  
 SYSTem:SECurity:IMMediate  
 SYSTem:TIME (?)  
 SYSTem:UPTime?  
 SYSTem:VERSion?

**Trigger commands**

ABORt  
 \*TRG  
 TRIGger:SEQuence:IMMediate  
 TRIGger:SEQuence:IMPedance (?)  
 TRIGger:SEQuence:LEVel (?)  
 TRIGger:SEQuence:POLarity (?)  
 TRIGger:SEQuence:SLOPe (?)  
 TRIGger:SEQuence:SOURce (?)  
 TRIGger:SEQuence:TIMer (?)

## Command Summaries

Tables 2–5 through 2–19 describe each command in each of the 12 functional groups.

### AWG Control Commands

The AWG Control commands control operating modes. This command group is not SCPI approved.

**Table 2–5: AWG Control commands**

Header	Description
AWGControl:CLOCK:SOURce(?)	Select the clock source (AWG400/500 series only)
AWGControl:DOUTput<x> [:STATe] (?)	Output the raw D/A converter output
AWGControl:ENHanced:SEQuence [:JMODE] (?)	Select the jump mode
AWGControl:EVENT[:LOGic] [:IMMEDIATE]	Generate the event signal for logic jump
AWGControl:EVENT:SOFTware [:IMMEDIATE] <line>	Jump to the specified line in the sequence file
AWGControl:EVENT:TABLE [:IMMEDIATE]	Generate the event signal for table jump
AWGControl:FG:FREQuency [:CW :FIXed] (?)	Set the frequency of the function waveform
AWGControl:FG<n>:FUNCTion [:SHAPE] (?)	Select the function or type of waveform (square wave, sine wave, etc.)
AWGControl:FG<n>:PHASe [:ADJusT] (?)	Set the phase of the function waveform (AWG420, AWG430 and AWG520 only)
AWGControl:FG<n>:POLarity (?)	Set the polarity of the function waveform
AWGControl:FG<n>:PULSe :DCYCLe (?)	Set the duty cycle of the pulse waveform
AWGControl:FG[:STATe] (?)	Turn the function generator mode on or off
AWGControl:FG<n>:VOLTage [:LEVeL][:IMMEDIATE] [:AMPLitude] (?)	Set the peak-to-peak voltage of the function waveform
AWGControl:FG<n>:VOLTage [:LEVeL][:IMMEDIATE]:OFFSet(?)	Set the offset voltage of the function waveform
AWGControl:RMODE (?)	Select the run mode, such as triggered or gated
AWGControl:RSTate?	Query the current running status
AWGControl:RUN[:IMMEDIATE]	Enable the output
AWGControl:SREStore	Restore the settings from the specified file

**Table 2-5: AWG Control commands (Cont.)**

Header	Description
AWGControl:SSAVE	Store the settings to the specified file
AWGControl:STOP[:IMMEDIATE]	Stop the output

**Calibration Commands**

The Calibration commands calibrate the waveform generator.

**Table 2-6: Calibration commands**

Header	Description
*CAL?	Perform calibration
CALibration[:ALL] (?)	Perform calibration

**Diagnostic Commands**

The Diagnostic commands control self-test diagnostic routines.

**Table 2-7: Diagnostic commands**

Header	Description
DIAGnostic:DATA?	Query results of self-test
DIAGnostic[:IMMEDIATE] (?)	Start the self-test
DIAGnostic:SElect (?)	Select the self-test routine
*TST?	Perform self-test

**Display Commands**

The Display commands mimic manipulation of front-panel controls and set the presentation of textual information on the front panel display.

**Table 2-8: Display commands**

Header	Description
ABSTouch	Perform the function corresponding to the front-panel control selected
DISPlay:BRIGhtness (?)	Control brightness of the display (AWG500/600 series only)
DISPlay:ENABle (?)	Control ON/OFF of the display (AWG400 series only)
DISPlay:HILight:COLor (?)	Control hilight of the display (AWG400 series only)

## Hardcopy Commands

The Hardcopy commands are used to print the entire display to a specified file rather than printing to an external device.

The hardcopy commands used in this application do not conform to the 1999 and 1995 SCPI hardcopy standard. (The SCPI standards state that the `MMEemory:OPEN` and `MMEemory:CLOSe` commands are used to open and close the file specified by `MMEemory:NAME`, to accommodate feeding data from the `HCOPY` subsystem. This state-dependent style of feeding data is not used in the waveform generator.) Instead, the hardcopy commands are implemented in a way that more closely resembles previous waveform generator usage. The waveform generator implements the hardcopy commands as illustrated in the following example:

```
MMEemory:NAME "SAMPLE1.BMP"
MMEemory:OPEN
HCOPY:DESTination "MMEM"
HCOPY
MMEM:CLOSe
```

The above command sequence can be written as follows for the waveform generator:

```
MMEemory:NAME "SAMPLE1.BMP"
HCOPY
```

In this case, the entire display will be written to the `SAMPLE1.BMP` file.

**Table 2–9: Hardcopy commands**

Header	Description
<code>HCOPY:DESTination</code>	Set the destination
<code>HCOPY:DEvice:COLor (?)</code>	Select the color, or monochrome (AWG400 series only)
<code>HCOPY:DEvice:LANGuage (?)</code>	Select the data format
<code>HCOPY[:IMMediate]</code>	Initiate the plot, or print immediately
<code>HCOPY:SDUMp[:IMMediate]</code>	Plot or print the whole display

**Mass Memory Commands**

The Mass Memory commands provide mass storage capabilities.

**Selecting Mass Memory Devices.** The waveform generator supports the devices listed below. The network drives can be specified with the SYSTem command group.

**Table 2-10: Mass storage in AWG400/500/600 series**

String argument	Description
MAIN	Internal hard disk drive
FLOP or FLOPPY	Internal floppy disk drive
NET1	Network drive 1
NET2	Network drive 2
NET3	Network drive 3

**File Names.** The <file\_name> parameter is described in some Mass Memory commands with a string. The content of the string depends on the format needs of the mass storage media. In particular, the file name may contain characters for specifying subdirectories (e.g. “/”) and the period separator (“.”). The instrument checks the file format when reading, and processes the file based on its content, regardless of the file extension.

**Table 2-11: Mass Memory commands**

Header	Description
MMEemory:CATalog?	Query information on the mass storage media
MMEemory:CDIRectory (?)	Change the default directory for a file system
MMEemory:CLOSe	Close the file specified in NAME
MMEemory:COpy	Copy an existing file to a new file
MMEemory:DATA (?)	Load data into the file
MMEemory:DELeTe	Remove a file
MMEemory:FEED (?)	Feed data into the file specified in NAME
MMEemory:INITialize	Initialize the specified mass storage
MMEemory:MDIRectory	Make a directory
MMEemory:MOVE	Move an existing file to another file
MMEemory:MSIS (?)	Select the current mass storage
MMEemory:NAME (?)	Set the file name to be opened or closed
MMEemory:OPEN	Open the file specified in NAME

**Output Commands** The Output commands control the characteristics of the waveform output port. In Table 2–13, OUTPut<x> refers to the waveform output channel, where <x> represents a related channel number shown in Table 2–12.

**Table 2–12: Output channel**

Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
OUTPut1	CH1 analog signal output	○	○	○	○	○	○
OUTPut2	CH2 analog signal output	X	○	○	X	○	X
OUTPut3	CH3 analog signal output	X	X	○	X	X	X
OUTPut4	CH1 digital data output	Op03 only	Op03 only	Op03 only	X	X	X
OUTPut5	CH2 digital data output	X	Op03 only	Op03 only			
	digital data output				Op03 only	Op03 only	X
OUTPut6	CH3 digital data output	X	X	Op03 only	X	X	X
OUTPut7	Noise ADD signal	X	X	X	○	○	X

○:usable  
 X:not used  
 OpXX:usable OptXX only

**Table 2–13: Output commands**

Header	Description
OUTPut<x>:FILTer[:LPASs] :FREQuency (?)	Determine the cutoff frequency of the low pass filter
OUTPut<x>[:STATe] (?)	Control whether the output terminal is open or closed
OUTPut<x>:ISTate (?)	Set the inverted output on or off

**Source Commands** The Source commands set waveform and marker output parameters, such as frequency and level. SOURce<x> and MARKer<y> in these commands have the meanings shown in Table 2–14.

**Table 2–14: Available source and markers**

Symbol	Meaning	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
SOURce1	CH1 signal of waveform generator	O	O	O	O	O	O
SOURce2	CH2 signal of waveform generator	X	O	O	X	O	X
SOURce3	CH3 signal of waveform generator	X	X	O	X	X	X
SOURce4	CH1 digital data	Op03 only	Op03 only	Op03 only	X	X	X
SOURce5	CH2 digital data	X	Op03 only	Op03 only	Op03 only	Op03 only	X
SOURce6	CH3 digital data	X	X	Op03 only	X	X	X
SOURce7	CH1 noise ADD signal	O	O	O			
	Noise ADD signal				O	O	X
SOURce8	CH1 Ext. ADD signal	O	O	O	O	O	X
SOURce9	CH2 noise ADD signal	X	O	O	X	X	X
SOURce10	CH2 Ext. ADD signal	X	O	O	X	X	X
SOURce11	CH3 noise ADD signal	X	X	O	X	X	X
SOURce12	CH3 Ext. ADD signal	X	X	O	X	X	X
Marker1	Marker 1 signal	X	X	X	O	O	O
Marker2	Marker 2 signal	X	X	X	O	O	O

O:usable  
X:not used  
OpXX:usable OptXX only

**Table 2-15: Source commands**

Header	Description
[SOURce<x>]:COMBine:FEED (?)	Add or release noise or external signal to the output (AWG400/500 series only)
[SOURce<x>]:FREQuency[:CW]:FIXed (?)	Set sampling frequency for outputting waveform
[SOURce<x>]:FUNCTion:USER (?)	Specify the user-defined waveform or pattern file
[SOURce<x>]:MARKer[1 2]:DELay (?)	Set the marker delay relative to waveform output (AWG400/500 series only)
[SOURce<x>]:MARKer[1 2][:LEVEL][:IMMediate]:HIGH (?)	Set high level for marker output (AWG400/500 series only)
[SOURce<x>]:MARKer[1 2][:LEVEL][:IMMediate]:LOW (?)	Set low level for marker output (AWG400/500 series only)
SOURce<x>:POWer[:LEVe1][:IMMediate][:AMPLitude] (?)	Set the level for the noise generator output (AWG400/500 series only)
[SOURce<x>]:ROSCillator:SOURce (?)	Select the reference oscillator source
SOURce<x>:SKEW (?)	Set the SKEW (AWG400 series only)
[SOURce<x>]:VOLTagE[:LEVe1][:IMMediate][:AMPLitude] (?)	Set the actual magnitude of the output signal
SOURce5:VOLTagE[:LEVe1][:IMMediate]:HIGH (?)	Set the high level of a digital pattern signal (AWG500 series Opt 03 only)
SOURce5:VOLTagE[:LEVe1][:IMMediate]:LOW (?)	Set the low level of a digital pattern signal (AWG500 series Opt 03 only)
[SOURce<x>]:VOLTagE[:LEVe1][:IMMediate]:OFFSet (?)	Set the offset that is added to the output signal



**Status Commands**

The external controller uses the Status commands to coordinate operation between the waveform generator and other devices on the bus. The Status commands set and query the registers/queues of the waveform generator event/status reporting system. For more information about the registers and queues described in Table 2–16, refer to the *Status and Event Reporting* section on page 3–1.

**Table 2–16: Status commands**

Header	Description
*CLS	Clear all the event registers and queues
*ESE (?)	Set and query ESER
*ESR?	Query SESR
*PSC (?)	Set power-on status clear flag
*SRE (?)	Set and query SRER
STATus:OPERation:CONDition?	Query the contents of OCR
STATus:OPERation:ENABle (?)	Set the enable mask of OENR
STATus:OPERation[:EVENT]?	Query the contents of OEVR
STATus:PRESet	Preset OENR and QENR
STATus:QUESTionable:CONDition?	Query the contents of QCR
STATus:QUESTionable:ENABle (?)	Set the enable mask of QENR
STATus:QUESTionable[:EVENT]?	Query the contents of QEVR
*STB?	Query SBR

**Synchronization Commands**

The external controller uses the Synchronization commands to prevent external communications from interfering with waveform generator operation.

**Table 2–17: Synchronization commands**

Header	Description
*OPC (?)	Generate or return the operation complete message
*WAI	Hold off all commands until all pending operations complete

**System Commands**

The System commands control miscellaneous instrument functions, such as LAN communication, security, and time.

**Table 2-18: System commands**

Header	Description
*IDN?	Query ID information about the waveform generator
*OPT?	Query installed options
*RST	Reset the waveform generator
SYSTem:BEEPer[:IMMediate]	Generate an audible tone
SYSTem:COMMunicate:LAN:DHCP[:CLient][:LEASe:TIME] (?)	Control the lease time of DHCP client function (AWG400 series only)
SYSTem:COMMunicate:LAN:DHCP[:CLient]:STATe] (?)	Control the DHCP client function (AWG400 series only)
SYSTem:COMMunicate:LAN:FTP[:SERVer][:STATe] (?)	Control the FTP server function
SYSTem:COMMunicate:LAN:FTP[:SERVer]:VERSion (?)	Change the FTP version
SYSTem:COMMunicate:LAN:GATeway:ADDRess (?)	Set IP address of the gateway
SYSTem:COMMunicate:LAN:NFS:TLIMit (?)	Set the timeout of NFS
SYSTem:COMMunicate:LAN:PING?	Execute PING test for the specified IP address
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:ADDRess (?)	Set IP address of the remote host
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:FSYSem (?)	Set the mount directory of the remote host
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:NAME (?)	Set the name of the remote host
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:PROTocol (?)	Set the protocol of the communication between the waveform generator and the remote host
SYSTem:COMMunicate:LAN:RDEVice[1 2 3]:STATe] (?)	Control whether the communication with the specified remote host is enabled
SYSTem:COMMunicate:LAN[:SELF]:ADDRess (?)	Set IP address of the waveform generator
SYSTem:COMMunicate:LAN[:SELF]:MADDRess ?	Query the MAC address of the waveform generator (AWG400 series only)
SYSTem:COMMunicate:LAN[:SELF]:SMASK (?)	Set the subnet mask of the waveform generator
SYSTem:DATE (?)	Set the internal calendar

**Table 2-18: System commands (Cont.)**

<b>Header</b>	<b>Description</b>
SYSTem:ERRor[:NEXT]?	Query the next entry from the waveform generator's error/event queue
SYSTem:KDIRection (?)	Set the direction of cursor movement controlled by the general purpose knob
SYSTem:KEYBoard[:TYPE] (?)	Select the keyboard type
SYSTem:KLOCK (?)	Lock the front panel and keyboard
SYSTem:SECurity:IMMediate	Destroy all data and settings for security
SYSTem:TIME (?)	Set the internal clock
SYSTem:UPTime?	Query elapsed time from the power-on
SYSTem:VERSion?	Query the SCPI version number

**Trigger Commands**     The Trigger commands synchronize the waveform generator actions with events.

**Table 2-19: Trigger commands**

Header	Description
ABORt	Reset the trigger system
*TRG	Generate the trigger event
TRIGger[:SEquence][:IMMediate]	Immediately trigger the sequence operation
TRIGger[:SEquence]:IMPedance (?)	Select the input impedance of the external trigger
TRIGger[:SEquence]:LEVEl (?)	Set the trigger level
TRIGger[:SEquence]:POLarity (?)	Select the polarity of the trigger signal
TRIGger[:SEquence]:SLOPe (?)	Select the slope of the trigger signal
TRIGger[:SEquence]:SOURce (?)	Select the source for the event detector
TRIGger[:SEquence]:TIMer (?)	Set the period of the internal clock

# Command Descriptions

This subsection lists each command and query in the waveform generator command set in alphabetical order. Each command entry includes a command description and command group, related commands (if any), syntax, and arguments. Each entry also includes one or more usage examples.

This subsection fully spells out headers, mnemonics, and arguments with the minimal spelling shown in upper case. For example, to use the abbreviated version of the SOURce:FREQuency command, just type SOUR:FREQ.

The symbol “(?)” follows the command header of commands that can be used as either a command or a query; the symbol “?” follows commands that can only be used as a query. Commands that are command-only or query-only are noted as such.

## ABORt (No Query Form)

This command resets the trigger system and places all trigger sequences in the idle state. This command is equivalent to depressing the FORCE TRIGGER button on the front panel in the gated mode.

**Group** Trigger

**Related Commands** TRIGger[:SEquence] [:IMMediate], \*TRG

**Syntax** ABORt



**Arguments** None

**Examples** ABORt  
resets the trigger system.

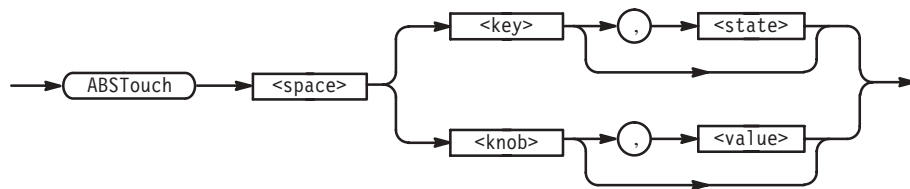
## ABSTouch (No Query Form)

This command performs the functions that are manually set by pressing the corresponding front-panel key and button, or by rotating the corresponding knob. This command works even when the instrument is in the keylock or local lockout states.

**Group** Display

**Related Commands** None

**Syntax** ABSTouch <key>[,<state>]  
 ABSTouch <knob>[,<value>]



**Arguments** <key> ::= BOTTom[1] | BOTTom2 | BOTTom3 | BOTTom4 | BOTTom5 | BOTTom6 | BOTTom7 | SIDE[1] | SIDE2 | SIDE3 | SIDE4 | SIDE5 | CMENu | RUN | CH1 | CH2 | CH3 | DIGita1 | DARRow | UARRow | LARRow | RARRow | SETup | APPL | EDIT | UTILity | HARDcopy | TOGGle | SHIFt | ENTEr | VMENu | QKEDit | HMENu | TMENu | FTTrigger | FEVEnt | SEVEnt | MEGa | EIGHT | KILO | NINE | MILLi | FOUR | MICRo | FIVE | NANO | SIX | PICO | ONE | D | TWO | E | THREe | F | ZERo | A | POINT | B | SIGN | C | CLR | G | DELEte | INF | RETurn | OUTPut[1] | IOUtpuT[1] | OUTPut2 | IOUtpuT2 | OUTPut3 | IOUtpuT3

**Table 2-20: Argument and available AWG**

Argument	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
BOTTom[1]	O	O	O	O	O	O
BOTTom2	O	O	O	O	O	O
BOTTom3	O	O	O	O	O	O
BOTTom4	O	O	O	O	O	O
BOTTom5	O	O	O	O	O	O
BOTTom6	O	O	O	O	O	O
BOTTom7	O	O	O	O	O	O
SIDE[1]	O	O	O	O	O	O

**Table 2-20: Argument and available AWG (Cont.)**

Argument	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
SIDe2	O	O	O	O	O	O
SIDe3	O	O	O	O	O	O
SIDe4	O	O	O	O	O	O
SIDe5	O	O	O	O	O	O
CMENu	O	O	O	O	O	O
RUN	O	O	O	O	O	O
CH1	O	O	O	O	O	X
CH2	X	O	O	X	O	X
CH3	X	X	O	X	X	X
DIGital	Op03	Op03	Op03	Op03	Op03	X
DARRow	O	O	O	O	O	O
UARRow	O	O	O	O	O	O
LARRow	O	O	O	O	O	O
RARRow	O	O	O	O	O	O
SETup	O	O	O	O	O	O
APPL	O	O	O	O	O	O
EDIT	O	O	O	O	O	O
UTILity	O	O	O	O	O	O
HARDcopy	O	O	O	O	O	O
TOGGle	O	O	O	O	O	O
SHIFt	O	O	O	O	O	O
ENTer	O	O	O	O	O	O
VMENu	O	O	O	O	O	O
QKEDit	O	O	O	O	O	O
HMENu	O	O	O	O	O	O
TMENu	O	O	O	O	O	O
FTRigger	O	O	O	O	O	O
FEVent	O	O	O	O	O	O
SEVen	O	O	O	O	O	O
MEGa	O	O	O	O	O	O
EIGHt	O	O	O	O	O	O
KILo	O	O	O	O	O	O
NINe	O	O	O	O	O	O
MILLi	O	O	O	O	O	O
FOUR	O	O	O	O	O	O
MICRo	O	O	O	O	O	O
FIVe	O	O	O	O	O	O

**Table 2-20: Argument and available AWG (Cont.)**

Argument	AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
NANo	O	O	O	O	O	O
SIX	O	O	O	O	O	O
PICo	O	O	O	O	O	O
ONE	O	O	O	O	O	O
D	O	O	O	O	O	O
TWO	O	O	O	O	O	O
E	O	O	O	O	O	O
THRee	O	O	O	O	O	O
F	O	O	O	O	O	O
ZERo	O	O	O	O	O	O
A	O	O	O	O	O	O
POINt	O	O	O	O	O	O
B	O	O	O	O	O	O
SIGN	O	O	O	O	O	O
C	O	O	O	O	O	O
CLR	O	O	O	O	O	O
G	O	O	O	O	O	O
DELeTe	O	O	O	O	O	O
INF	O	O	O	O	O	O
RETurn	O	O	O	O	O	O
OUTPut[1]	O	O	O	O	O	O
IOUtput[1]	except Op05	except Op05	except Op05	O	X	O
OUTPut2	X	O	O	X	O	X
IOUtput2	X	except Op05	except Op05	X	X	X
OUTPut3	X	X	O	X	X	X
IOUtput3	X	X	except Op05	X	X	X

O:usable

X:not used

OptXX:usable OptXX only

<knob>::= OFFSet | LSCale | HSHift | SSCale | LEVe1 | GPKNob

<state>::= ON | OFF | <NR1>

This argument sets the press and release of the specified front panel key. If you specify ON or nonzero value in this argument, the front panel key is set to press. If you specify OFF or zero value in this argument, the front panel key is set to release. When the argument is not specified, 1 is set.

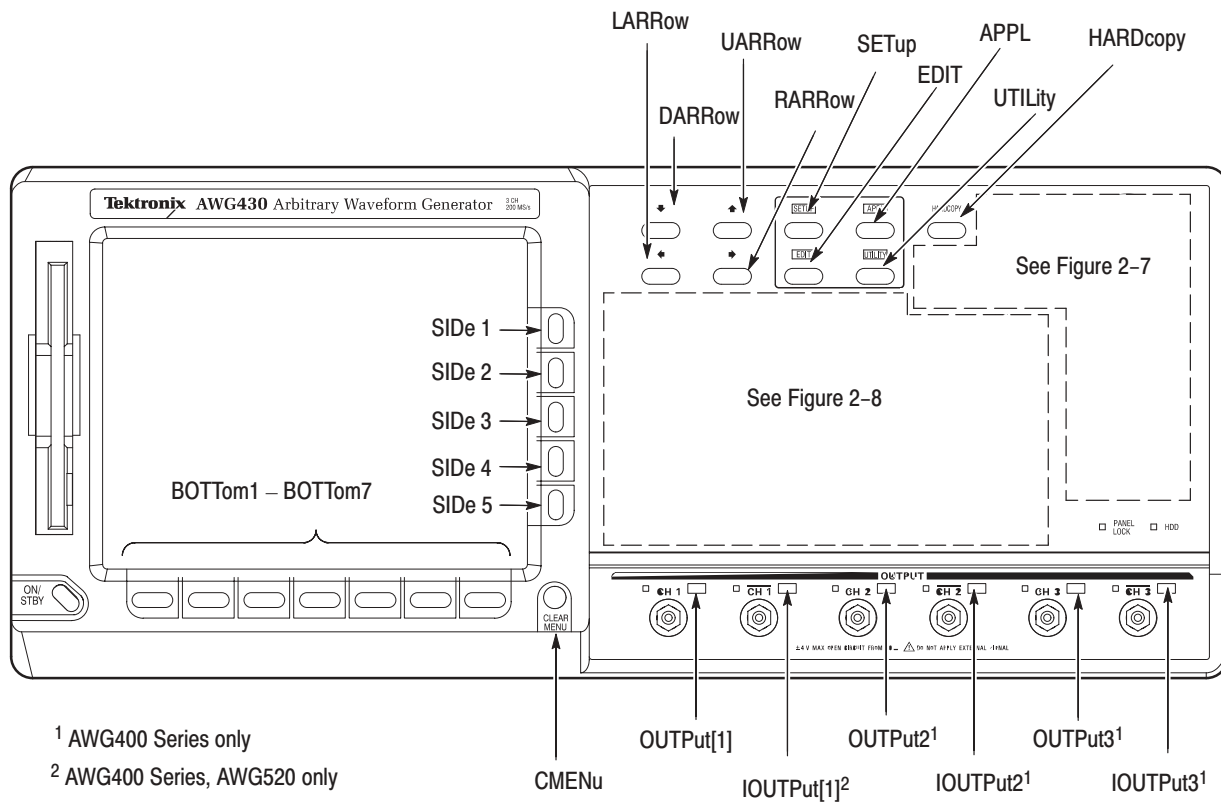
<value>::= <NR1>

This argument sets the rotating direction and quantities of the specified front



panel knob. If you specify a positive value in this argument, the knob rotates clockwise. If you specify a negative value in this argument, the knob rotates counterclockwise. When the argument is not specified, 1 is set.

Figure 2–6 shows ABSTouch arguments corresponding to the associated controls.



**Figure 2–6: ABSTouch arguments and associated controls**

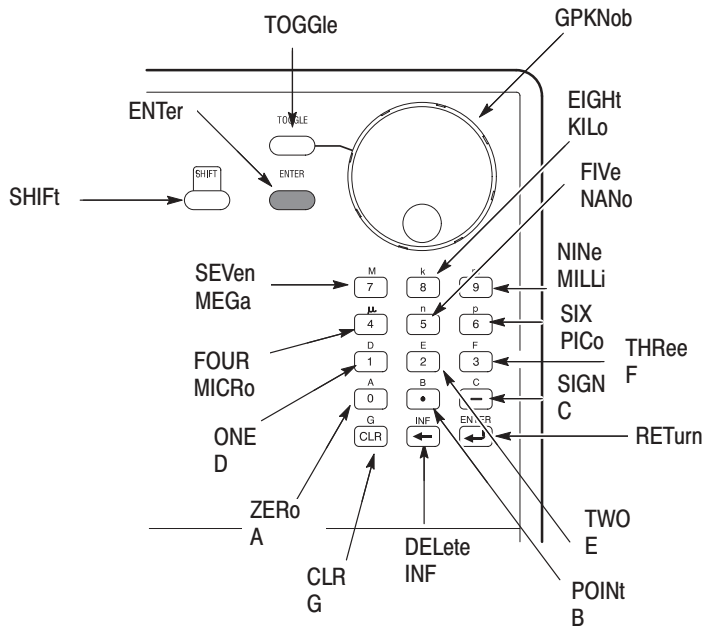
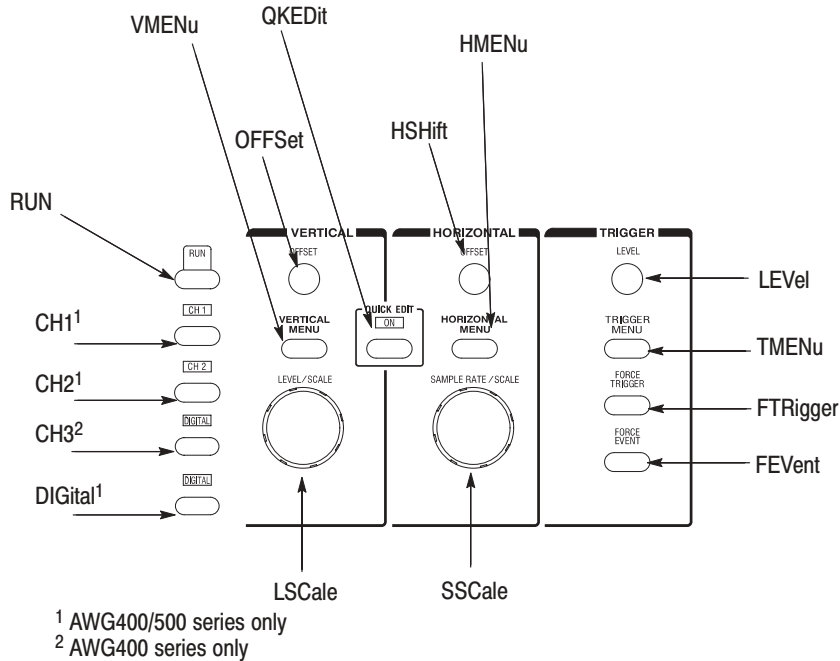


Figure 2-7: ABSTouch arguments and Front panel key area



<sup>1</sup> AWG400/500 series only  
<sup>2</sup> AWG400 series only

Figure 2-8: ABSTouch arguments and Trigger, Output controls

**Examples**    `ABSTOUCH SETUP`  
displays the setup menu that is displayed by pressing the SETUP button on the front panel.

## AWGControl:CLOCK:SOURce (?)

*AWG400/500 series Only*

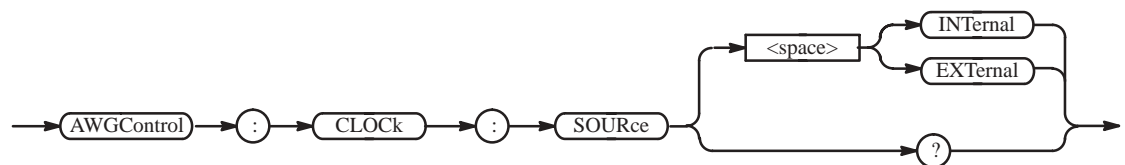
This command selects the clock source as either Internal or External.

**Group**    AWG Control

**Related Commands**    `SOURce:ROSCillator:SOURce`

**Syntax**    `AWGControl:CLOCK:SOURce { INTernal | EXTernal }`

`AWGControl:CLOCK:SOURce?`



**Arguments**    `INTernal` select the internal clock derived from the reference clock as the clock source.

`EXTernal` select the external clock signal connected to the EXT CLOCK IN on the rear panel.

At \*RST, this parameter is set to `INTernal`.

**Examples**    `AWGControl:CLOCK:SOURce EXTernal`  
selects the external clock as the clock source.

## AWGControl:DOUOutput<x>[:STATe] (?)

This command supplies raw output of the waveform generator D/A converter for the specified channel. In AWG500 series the settings of `SOURce:VOLTage` and

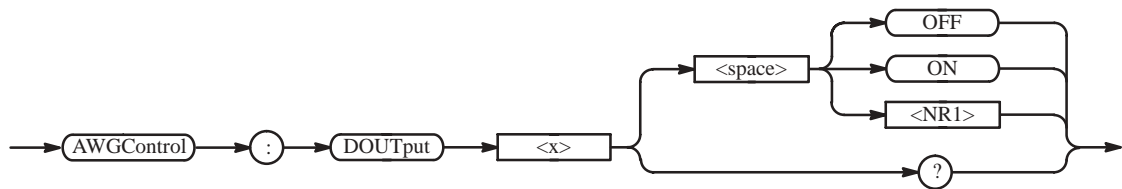
OUTPut:FILTER commands are ignored. In the AWG600 series, the setting OUTPut:FILTER command is ignored.

**Group** AWG Control

**Related Commands** SOURce:VOLTage command group, OUTPut:FILTER command group

**Syntax** AWGControl:DOUTput<x>[:STATE] { OFF | ON | <NR1> }  
 AWGControl:DOUTput<x>[:STATE]?

AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
1	1, 2	1, 2, 3	1	1, 2	1



**Arguments** OFF or <NR1> = 0 provides the D/A converter output normally.  
 ON or <NR1> ≠ 0 provides raw output of the D/A converter.  
 At \*RST, this value is set to 0.

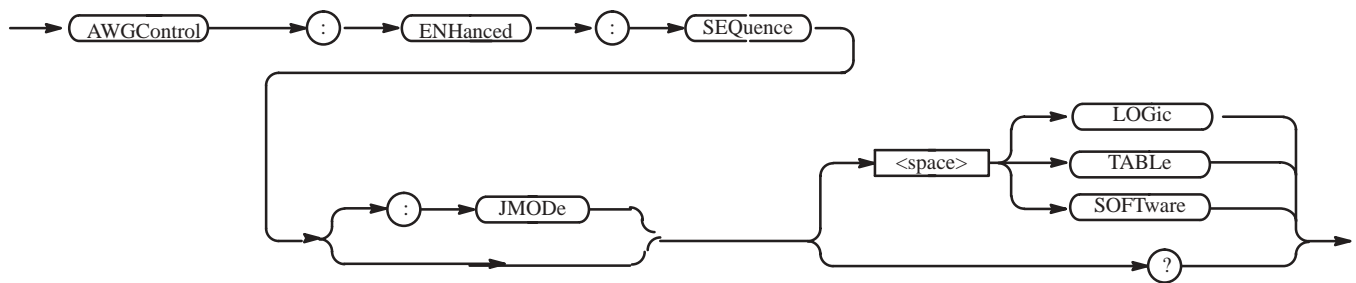
**Examples** AWGControl:DOUTput1:STATE ON  
 supplies the D/A converter output directly to CH 1.

## AWGControl:ENHanced:SEquence[:JMODe] (?)

This command selects the jump mode in the sequence of the enhanced mode.

**Group** AWG Control

**Syntax** AWGcontrol:ENHanced:SEquence[:JMODe] {LOGic | TABLe | SOFTware}  
AWGControl:ENHanced:SEquence[:JMODe]?



**Arguments**

LOGic	The jump mode is "logic".
TABLe	The jump mode is "table".
SOFTware	The jump mode is "software".

At \*RST, this value is set to TABLe.

**Examples** AWGControl:ENHanced:SEquence SOFTWARE  
sets the jump mode to software.

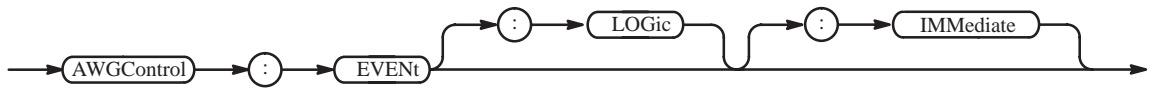
## AWGControl:EVENT[:LOGic][:IMMEDIATE] (No Query Form)

This command generates a trigger event for the "logic jump" specified in the sequence file. This has the same effect as pressing the FORCE EVENT button on the front panel.

**Group** AWG Control

**Related Commands** AWGControl:RUN[:IMMEDIATE], \*TRG

**Syntax** AWGControl:EVENT[:LOGic][:IMMEDIATE]



**Arguments** None

**Examples** `AWGControl:EVENT:LOGic:IMMediate`  
 generates a trigger event for the “logic jump”.

### **AWGControl:EVENT:SOFTware[:IMMediate] (No Query Form)**

This command jumps to a specified line in a sequence file. To enable this command, a sequence file must be loaded and software jump mode must be set in the sequence file.

This command will return a “Settings conflict” error (code:–221) when any of these conditions are present:

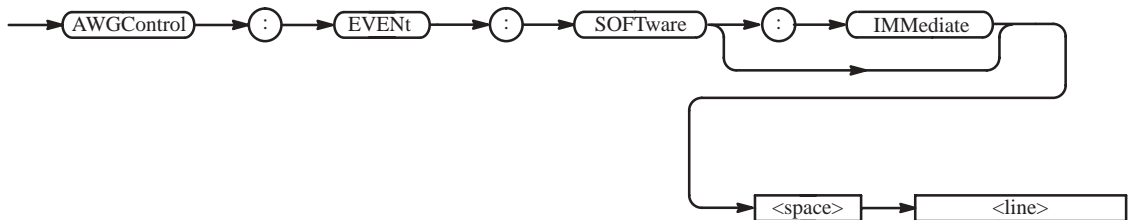
- 1) The waveform generator is not in Enhanced mode.
- 2) No sequence file is loaded.
- 3) The Jump Mode setting of the sequence file is not Software.

It also will return a “Data out of range” error (code:–222) if the <line> argument is less than or equal to zero, or greater than the number of steps of the loaded sequence file.

**Group** AWG Control

**Related Commands** None

**Syntax** `AWGControl:EVENT:SOFTware[:IMMediate] <line>`



**Arguments** `<line>::=<NR1>` is the line number to be jumped to in the sequence file.

**Examples** `AWGControl:EVENT:SOFTWARE:IMMEDIATE 10`  
jumps to line 10 in the sequence file.

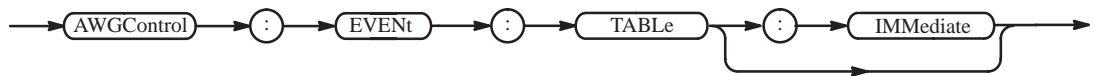
## AWGControl:EVENT:TABLE[:IMMEDIATE] (No Query Form)

This command generates a trigger event for the “table jump” specified in the sequence file when a jump mode of sequence (ENHanced mode) is a Table.

**Group** AWG Control

**Related Commands** None

**Syntax** `AWGControl:EVENT:TABLE[:IMMEDIATE]`



**Arguments** None

**Examples** `AWGControl:EVENT:TABLE:IMMEDIATE`  
generates a trigger event for the “table jump”.

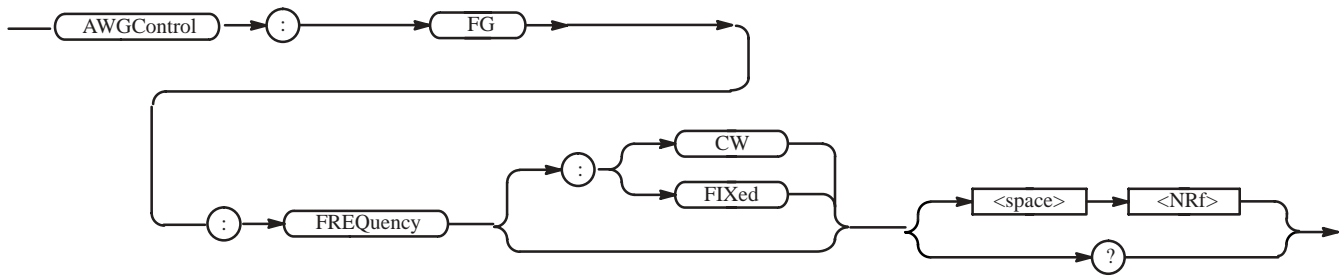
## AWGControl:FG:FREQUENCY[:CW|:FIXED] (?)

This command adjusts the frequency of the function waveform on all channels. This query returns the frequency currently set.

CW (Continuous Wave) and FIXED are aliases, and have the same effect.

**Group** AWG Control

**Syntax** `AWGControl:FG:FREQUENCY[:CW|:FIXED] <NRf>`  
`AWGControl:FG:FREQUENCY[:CW|:FIXED]?`



**Arguments** <NRf> is the output waveform frequency. The range is as follows.

- AWG400 series : 1Hz to 10 MHz.
- AWG500 series : 1Hz to 100 MHz.
- AWG600 series : 1Hz to 260 MHz.

At \*RST, this value is set as follows.

- AWG400 series : 1 MHz.
- AWG500 series : 5 MHz.
- AWG600 series : 13 MHz.

**Examples** AWGControl:FG:FREQuency 10MHz  
sets the frequency to 10 MHz.

## AWGControl:FG<n>:FUNcTion[:SHAPE] (?)

This command selects a standard function waveform (as opposed to a waveform file) in the specified channel.

This query returns the currently selected standard function waveform.

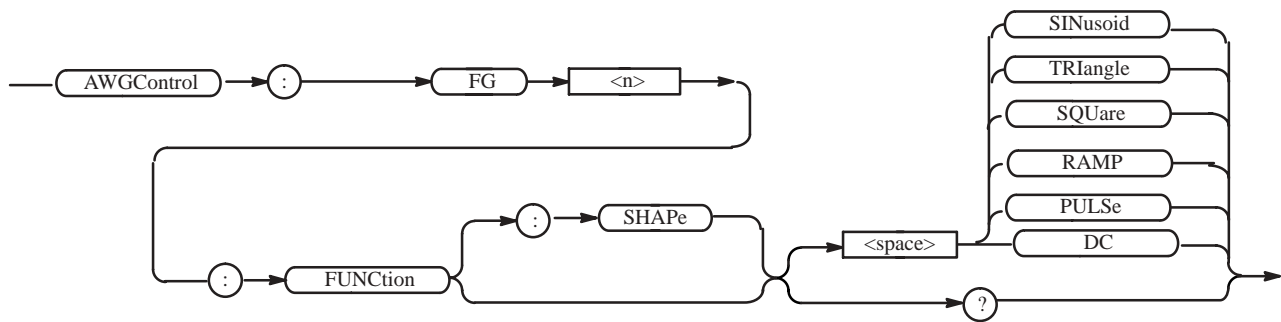
**Group** AWG Control

**Syntax** AWGControl:FG<n>:FUNcTion[:SHAPE] <shape>

AWGControl:FG<n>:FUNcTion[:SHAPE] ?

<n> ::= channel





**Arguments** SINusoid selects a sine wave function waveform.  
 TRIangle selects a triangle function waveform.  
 SQUare selects a square wave function waveform.  
 RAMP selects a ramp function waveform.  
 PULSe selects a pulse function waveform.  
 DC selects a DC function waveform.

At \*RST, this value is set to SINusoid

**Examples** AWGControl:FG1:FUNCTION RAMP  
 sets the CH1 standard function waveform to RAMP.

## AWGControl:FG<n>:PHASe[:ADJust] (?)

**AWG420, AWG430 and AWG520 Only**

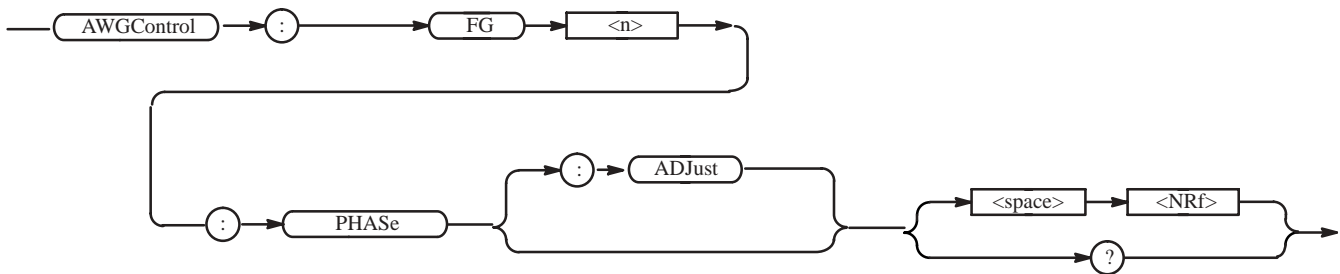
This command sets the phase of the function waveform on selected channel.  
This query returns the phase of the function waveform.

**Group** AWG Control

**Syntax** AWGControl:FG<n>:PHASe[:ADJust] <NRf>

AWGControl:FG<n>:PHASe[:ADJust]?

<n> ::= channel



**Arguments** <NRf> is the phase of the function waveform. The range is  $-6.28319$  ( $-360$ degrees) to  $+6.28319$  ( $+360$ degrees).

Step:

AWG400	Frequency	Step(degree)
	1.000Hz to 20.00kHz	0.036
	20.01kHz to 200.0kHz	0.36
	200.1kHz to 2.000MHz	3.6
	2.001MHz to 4.000MHz	7.2
	4.001MHz to 5.000MHz	9
	5.001MHz to 8.000MHz	14.4
	8.001MHz to 10.00MHz	18

AWG500	Frequency	Step(degree)
	1.000Hz to 100.0kHz	0.036
	100.1kHz to 1.000MHz	0.36
	1.001MHz to 5.000MHz	1.8
	5.001MHz to 10.00MHz	3.6
	10.01MHz to 20.00MHz	7.2
	20.01MHz to 25.00MHz	9
	25.01MHz to 40.00MHz	14.4
	40.01MHz to 50.00MHz	18
	50.01MHz to 100.0MHz	36

At \*RST, this value is set as 0.

**Examples** `AWGControl:FG1:PHASe 0.36DEGree`  
sets the CH1 phase to 0.36 degrees.

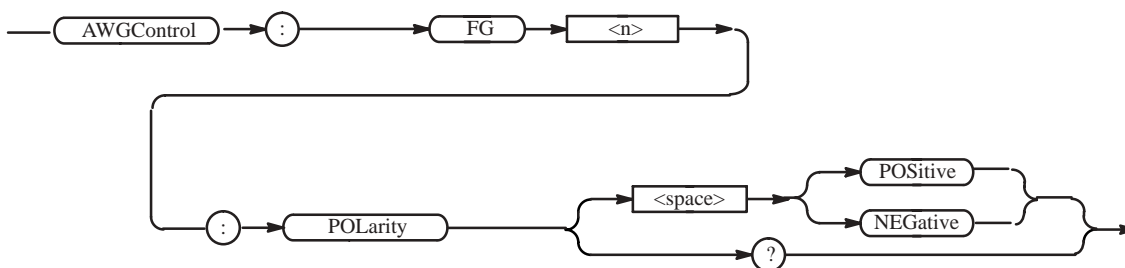
## AWGControl:FG<n>:POLarity (?)

This command sets polarity of the function waveform on the selected channel.  
This query returns polarity currently set.

**Group** AWG Control

**Syntax** `AWGControl:FG<n>:POLarity {POSitive | NEGative}`  
`AWGControl:FG<n>:POLarity?`

<n> ::= channel



**Arguments** POSitive sets waveform to positive polarity.  
NEGative sets waveform to negative polarity.

At \*RST, this value is set to POSitive.

**Examples** `AWGControl:FG1:POLarity POSitive`  
sets the CH1 polarity to positive.

## AWGControl:FG<n>:PULSe:DCYClE (?)

This command sets the duty cycle of the pulse waveform on selected channel.

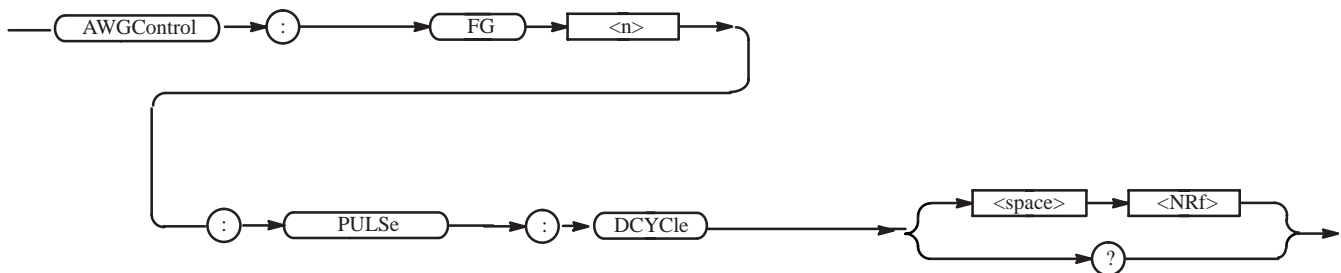
This query returns the duty cycle of the pulse waveform.

**Group** AWG Control

**Syntax** AWGControl:FG<n>:PULSe:DCYClE <NRf>

AWGControl:FG<n>:PULSe:DCYClE?

<n> ::= channel



**Arguments** <NRf> is the duty cycle. The range is 0.1 to 99.9%.

Step:

AWG400	Frequency	Step(%)
	1.000Hz to 200.0kHz	0.1
	200.1kHz to 2.000MHz	1
	2.001MHz to 4.000MHz	2
	4.001MHz to 5.000MHz	2.5
	5.001MHz to 8.000MHz	4
	8.001MHz to 10.00MHz	5

AWG500	Frequency	Step(%)
	1.000Hz to 1.000MHz	0.1
	1.001MHz to 5.000MHz	0.5
	5.001MHz to 10.00MHz	1
	10.01MHz to 20.00MHz	2
	20.01MHz to 25.00MHz	2.5
	25.01MHz to 40.00MHz	4
	40.01MHz to 50.00MHz	5
	50.01MHz to 100.0MHz	10

AWG600	Frequency	Step(%)
	1.000Hz to 2.600MHz	0.1
	2.601MHz to 13.00MHz	0.5
	13.01MHz to 26.00MHz	1
	26.01MHz to 52.00MHz	2

52.01MHz to 65.00MHz	2.5
65.01MHz to 104.0MHz	4
104.1MHz to 130.0MHz	5
130.1MHz to 260.0MHz	10

At \*RST, this value is set as 10.0.

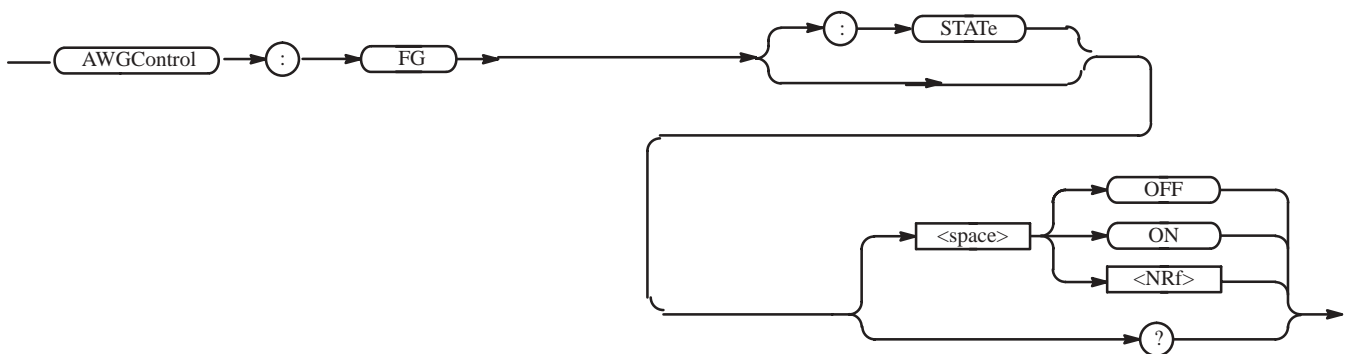
**Examples** `AWGControl:FG1:PULSe:DCYClE 20`  
sets the CH1 duty cycle to 20%.

## AWGControl:FG[:STATe] (?)

This command turns the FG(Function Generation) mode on or off.  
This query returns status indicating whether the waveform generator is set to the function generator mode.

**Group** AWG Control

**Syntax** `AWGControl:FG[:STATe] { OFF | ON | <NRf> }`  
`AWGControl:FG[:STATe]?`



**Arguments** OFF or `<NRf> = 0` sets the FG mode to OFF .  
ON or `<NRf> ≠ 0` sets the FG mode to ON.  
At \*RST, this value is set to OFF.

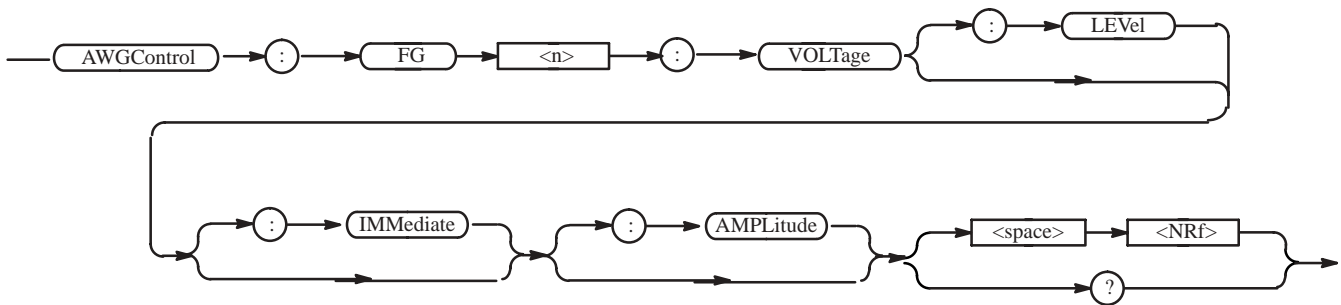
**Examples** `AWGControl:FG ON`  
sets the FG mode to ON.

## AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate][:AMPLitude] (?)

This command adjusts peak-to-peak voltage of the function waveform on selected channel.  
 This query returns peak-to-peak voltage currently set.

**Group** AWG Control

**Syntax** AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf>  
 AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate][:AMPLitude]?  
 <n> ::= channel



**Arguments** <NRf> is the amplitude of the waveform. The Step is 1mV.  
 The range is as follows.

- AWG400(Standard) series : 0.020Vpp to 2.000Vpp
- AWG400(Option 05) series : 0.020Vpp to 5.000Vpp
- AWG500 series : 0.020Vpp to 2.000Vpp
- AWG600 series : 0.020Vpp to 2.000Vpp

At \*RST, this value is set to 1.0.

**Examples** AWGControl:FG1:VOLTage 2.0  
 sets the CH1 amplitude to 2.000Vpp.

## AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate]:OFFSet (?)

This command adjusts offset voltage of the function waveform on selected channel.

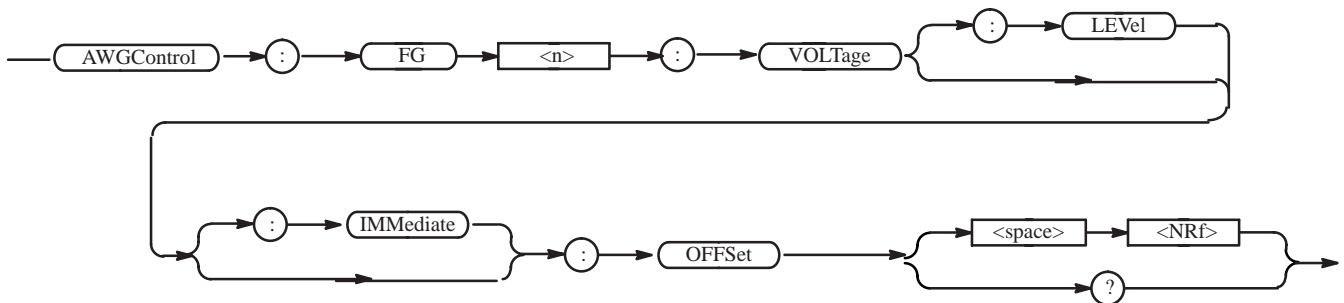
This query returns offset voltage currently set.

**Group** AWG Control

**Syntax** AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate]:OFFSet <NRf>

AWGControl:FG<n>:VOLTage[:LEVel][:IMMediate]:OFFSet?

<n> ::= channel



**Arguments** <NRf> is the offset of the waveform. The Step is 1mV.  
The range is as follows.

AWG400(Standard) series : -1.000V to +1.000V  
 AWG400(Option 05) series : -2.500V to +2.500V  
 AWG500 series : -1.000V to +1.000V  
 AWG600 series : -1.000V to +1.000V

At \*RST, this value is set to 0.0.

**Examples** AWGControl:FG1:VOLTage:OFFSet 0.5  
 sets the CH1 offset to 0.5V.

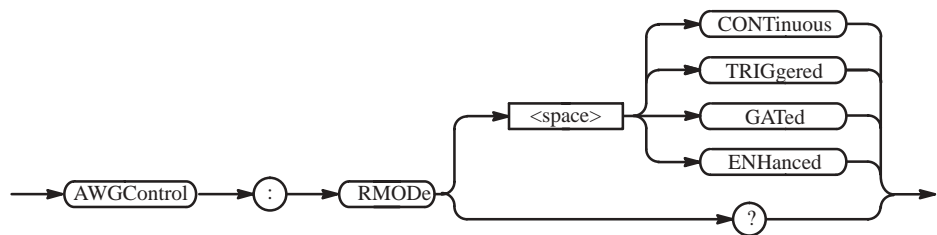
## AWGControl:RMODe (?)

This command selects the mode used to output waveforms or sequences.

**Group** AWG Control

**Related Commands** AWGControl:RUN[:IMMediate], AWGControl:STOP[:IMMediate], [SOURce[1]]:FUNct ion:USER, \*TRG

**Syntax** AWGControl:RMODe { CONTInuous | TRIGgered | GATed | ENHanced }  
AWGControl:RMODe?



**Arguments** You can select the modes listed in Table 2–21.

**Table 2–21: Selecting run modes**

Arguments	Descriptions
CONTInuous	Sets the continuous mode, which continuously outputs the waveform. The external trigger, including FORCE TRIGGER button and the corresponding remote commands, have no effect.
TRIGgered	Sets the triggered mode, which outputs one waveform cycle for each trigger.
GATed	Sets the gated mode, which continuously outputs the waveform or sequence as long as the trigger remains enabled. The trigger remains effective as long as any of the following events occur: <ul style="list-style-type: none"> <li>■ The FORCE TRIGGER button remains pressed</li> <li>■ A valid external gate signal remains input</li> <li>■ The TRIGger[:SEquence] [:IMMediate] or *TRG command has been executed but an ABORt command has not yet been issued</li> </ul>
ENHanced	Sets the enhanced mode, which outputs the waveform according to the sequence file specified with the SOURce:FUNct ion:USER command. If the sequence file is not loaded, this mode is the same as the triggered mode.

At \*RST, this parameter is set to CONTInuous.



**Examples** SOURce:FUNction:USER "SAMPLE1.SEQ";:AWGControl:RMODe ENHanced;RUN  
outputs waveform according to the sequence file SAMPLE1.SEQ.

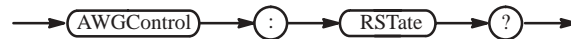
AWGControl:RMODe?  
can return the following response:  
TRIG

## AWGControl:RSTate? (Query Only)

This command returns the current running status.

**Group** AWG Control

**Syntax** AWGControl:RSTate?



**Arguments** None

**Returns** <NR1>

- 0 The waveform generator is stopped.
- 1 The waveform generator is waiting for a trigger.
- 2 The waveform generator is running.

**Examples** AWGControl:RSTate?  
can return the following response:  
1

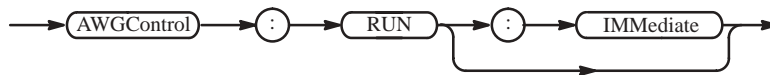
## AWGControl:RUN[:IMMEDIATE] (No Query Form)

This command initiates the output of a waveform or a sequence. This has the same effect as manually pressing the RUN button on the front panel.

**Group** AWG Control

**Related Commands** AWGControl:STOP[:IMMEDIATE], \*TRG

**Syntax** AWGControl:RUN[:IMMEDIATE]



**Arguments** None

**Examples** AWGControl:RUN[:IMMEDIATE]  
initiates the output of a waveform or a sequence.

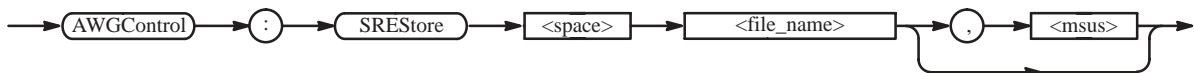
## AWGControl:SREStore (No Query Form)

This command resets the default settings of a specified file.

**Group** AWG Control

**Related Commands** AWGControl:SSAVE, MMEemory:CDIRECTORY, MMEemory:MSIS

**Syntax** AWGControl:SREStore <file\_name>[,<msus>]



**Arguments** <file\_name>::=<string> specifies the file to restore the settings.  
<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTEM:COMMunicate:LAN commands)

**Examples** AWGControl:SREStore "SAMPLE1.SET", "FLOppy"  
resets the default settings of the file SAMPLE1.SET on the floppy disk.

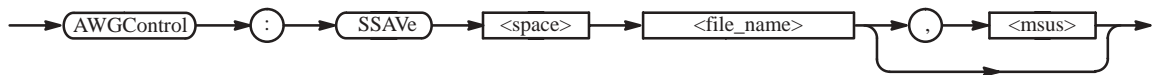
## AWGControl:SSAVe (No Query Form)

This command stores the current settings to a specified file.

**Group** AWG Control

**Related Commands** AWGControl:SREStore, MMEemory:CDIRectory, MMEemory:MSIS

**Syntax** AWGControl:SSAVe <file\_name>[,<msus>]



**Arguments** <file\_name>::=<string> specifies the file to store the settings.

<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTem:COMMUnicate:LAN commands)

**Examples** AWGControl:SSAVe "SAMPLE1.SET", "FLOppy"  
stores the current settings to the file SAMPLE1.SET on the floppy disk.

## AWGControl:STOP[:IMMediate] (No Query Form)

This command terminates waveform output. When the mode is not set to continuous, it also resets the sequence pointer to output the waveform from the top of the sequence with the next trigger event.

**Group** AWG Control

**Related Commands** AWGControl:RUN[:IMMediate], \*TRG

**Syntax** AWGControl:STOP[:IMMEDIATE]



**Arguments** None

**Examples** AWGControl:STOP[:IMMEDIATE]  
stops the output of a waveform.

## \*CAL? (Query Only)

The \*CAL? query performs an internal calibration and returns a status that indicates whether or not the waveform generator completed the calibration successfully. If an error is detected during calibration, execution immediately stops, and an error code is returned. This query performs the same function as the CALibration[:ALL]? query.

---

**NOTE.** A period of time is required to complete the internal calibration. During this time, the waveform generator does not respond to any commands or queries issued.

---

**Group** Calibration

**Related Commands** CALibration[:ALL]?

**Syntax** \*CAL?



**Arguments** None

**Returns** <NR1>

0 Terminated without error.  
-340 Calibration failed.

**Examples** \*CAL?  
performs an internal calibration and returns the results. For example, the query might return 0, which indicates the calibration terminated without any errors.

## CALibration[:ALL] (?)

The CALibration[:ALL] command performs a full calibration of the waveform generator.

The CALibration[:ALL]? query performs a full calibration and responds with a <NR1> indicating the success of the calibration. This query has the same function as the \*CAL? query.

If an error is detected during calibration, a message is queued in the error/event queue, and the error code “-340” is returned.

---

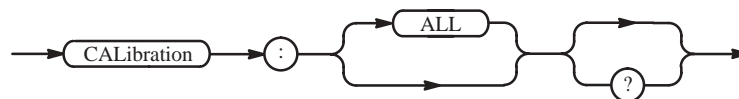
**NOTE.** A period of time is required to complete the internal calibration. During this time, the waveform generator does not respond to any commands or queries issued.

---

**Group** Calibration

**Related Commands** \*CAL?

**Syntax** CALibration[:ALL]  
CALibration[:ALL]?



**Arguments** None

**Returns** <NR1>  
0 Terminated without error.  
-340 Calibration failed.

**Examples** CALibration[:ALL]  
performs a full calibration.

CALibration[:ALL]?

performs a full calibration and returns the results. For example, it might return 0, which indicates the calibration terminated without any errors.

## \*CLS (No Query Form)

This command clears all the event registers and queues, used by the waveform generator status and event reporting system. For more details, refer to the, *Status and Events* section.

**Group** Status

**Syntax** \*CLS



**Arguments** None

**Examples** \*CLS  
clears all the event registers and queues.

## DIAGnostic:DATA? (Query Only)

This command returns the results of a self-test.

**Group** Diagnostic

**Related Commands** DIAGnostic[:IMMEDIATE], DIAGnostic:SElect

**Syntax** DIAGnostic:DATA?



**Arguments** None

**Returns** <NR1>

0 Terminated without error.  
 -330 Self-test failed.

**Examples** `DIAGnostic:DATA?`  
 might return 0.

## DIAGnostic[:IMMediate] (?)

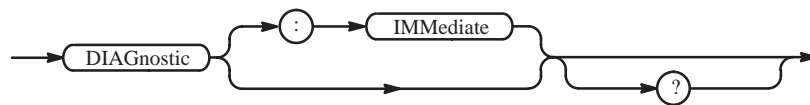
The `DIAGnostic[:IMMediate]` command executes the self-test routine(s) selected by the `DIAGnostic:SElect` command. The query `DIAGnostic[:IMMediate]?` executes the routine(s) and returns the results.

If an error is detected during execution, the routine that detected the error terminates. If all of the self-test routines are selected, self-testing continues with execution of the next self-test routine.

**Group** Diagnostic

**Related Commands** `DIAGnostic:SElect`, `DIAGnostic:DATA?`

**Syntax** `DIAGnostic[:IMMediate]`  
`DIAGnostic[:IMMediate]?`



**Arguments** None

**Returns** <NR1>

0 Terminated without error.  
 -330 Self-test failed.

**Examples** `DIAGnostic:SElect ALL;IMMediate?`  
 executes all of the self-test routines. After all self-test routines finish, the results of the self-tests are returned.

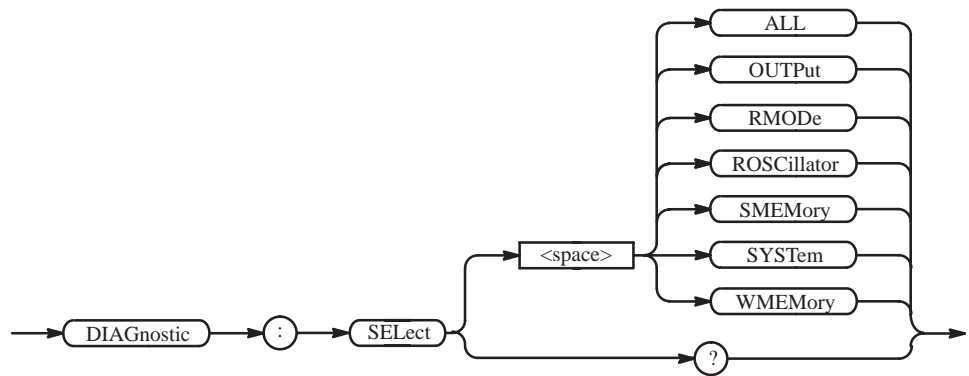
## DIAGnostic:SElect (?)

This command selects the self-test routine(s).

**Group** Diagnostic

**Related Commands** DIAGnostic[:IMMediate]

**Syntax** DIAGnostic:SElect { ALL | OUTPut | RMODe | ROSeillator  
| SMeMory | SYSTem | WMeMory }  
DIAGnostic:SElect?



**Arguments** You can select the following self-test routines:

**Table 2-22: Self-test routines**

Argument	Description
ALL	Checks all routines that follow
OUTput	Checks the analog output unit
RMODe	Checks the control unit
ROSeillator	Checks the reference oscillator unit
SMeMory	Checks the sequence memory
SYSTem	Checks the system unit, such as the system memory
WMeMory	Checks the waveform memory

At \*RST, this parameter is set to ALL.



**Examples**     `DIAGnostic:SElect WMEMory;IMMediate`  
executes the waveform memory self-test routine.

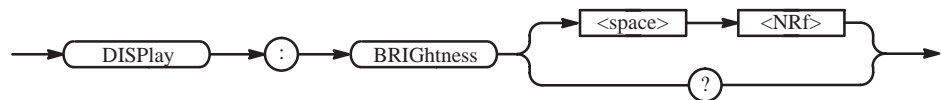
## DISPlay:BRIGhtness (?)

*AWG500/600 series Only*

This command controls the intensity of the display.

**Group**     Display

**Syntax**     `DISPlay:BRIGhtness <NR2>`  
`DISPlay:BRIGhtness?`



**Arguments**     `<NR2>` ranges from 0 to 1, where 1 is full intensity and 0 is fully blanked.  
At `*RST`, this value is set to 0.7.

**Examples**     `DISPlay:BRIGhtness 0.8`  
sets the intensity of the display to 80% of maximum intensity.

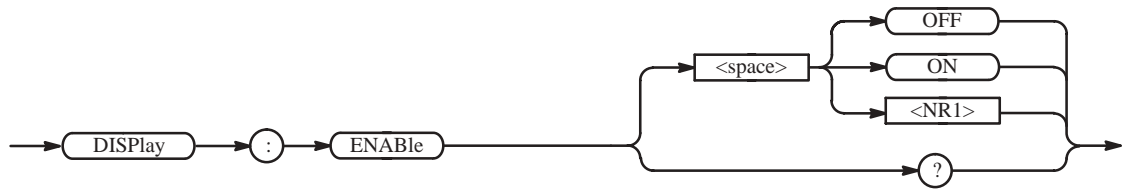
## DISPlay:ENABLE (?)

*AWG400 series Only*

This command controls ON/OFF of the display.

**Group**     Display

**Syntax**     `DISPlay:ENABLE { OFF | ON | <NR1> }`  
`DISPlay:ENABLE?`



**Arguments** OFF or <NR1> = 0 sets OFF the display.  
 ON or <NR1> ≠ 0 sets ON the display.  
 At \*RST, this value is set to OFF.

**Examples** DISPlay:ENABle ON  
 sets ON the display.

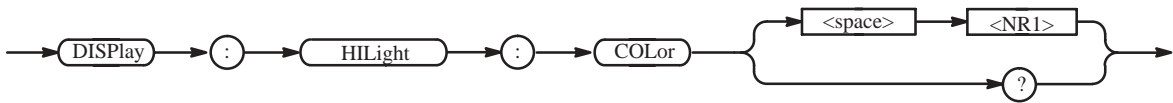
## DISPlay:HILight:COLor (?)

*AWG400 series Only*

This command controls the HILight color.

**Group** Display

**Syntax** DISPlay:HILight:COLor <NR1>  
 DISPlay:HILight:COLor?



**Arguments** <NR1> is the color number. The range is 0 to 7.  
 At \*RST, this value is set to 0.

**Returns** <NR1> indicates the number of color.

**Examples** DISPlay:HILight:COLor 1  
 sets the number 1 color.

**\*ESE (?)**

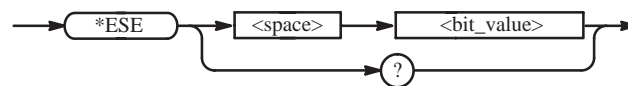
The \*ESE command sets the bits of the ESER (Event Status Enable Register) used in the status and events reporting system of the waveform generator. The \*ESE? query returns the contents of the ESER. Refer to the *Status and Events* for more information about the ESER.

**Group** Status

**Related Commands** \*CLS, \*ESR?, \*PSC, \*SRE, \*STB?

**Syntax** \*ESE <bit\_value>

\*ESE?

**Arguments**

<bit\_value> ::= <NR1>

where <NR1> is a decimal integer in the range 0 to 255. The binary bits of the ESER are set according to this value.

The power-on default for ESER is 0 if \*PSC is 1. If \*PSC is 0, the ESER maintains its value through a power cycle.

**Examples**

\*ESE 177

sets the ESER to 177 (binary 10110001), which sets the PON, CME, EXE and OPC bits.

\*ESE?

might return 176, which indicates that the ESER contains the binary number 10110000.

## \*ESR? (Query Only)

This command returns the contents of the Standard Event Status Register (SESR) used in the status and events reporting system in the waveform generator. \*ESR? also clears the SESR (since reading the SESR clears it). Refer to Section 3 *Status and Events* for more information.

**Group** Status

**Related Commands** \*CLS, \*ESE?, \*SRE, \*STB?

**Syntax** \*ESR?



**Returns** <NR1> indicates the content of the SESR in a decimal integer.

**Examples** \*ESR?  
might return 181, which indicates that the SESR contains the binary number 10110101.

## HCOPY:DESTination (No Query Form)

This command sets the hardcopy destination. For the waveform generator, the destination is always set to MMEMory (mass memory). This command is included only for compatibility with the SCPI standard. The destination file on the mass memory device is specified by the MMEMory:NAME command. For more information about hardcopy, see *Hardcopy Commands* on page 2–22.

**Group** Hardcopy

**Related Commands** MMEMory:NAME

**Syntax** HCOpy:DESTination <data\_handle>



**Arguments** <data\_handle>::=<string>  
where <string> is fixed to "MMEMory" for the waveform generator.

**Examples**    `HCOPY:DESTination "MMEMory"`  
sets the hardcopy destination to a file specified with the `MMEMory:NAME` command.

## HCOPY:DEvice:COLor (?)

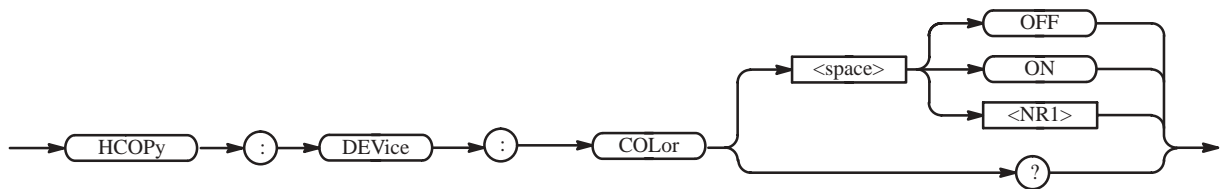
*AWG400 series Only*

This command sets the hardcopy color mode.

**Group**    Hardcopy

**Related Commands**    `HCOPY:DEvice:LANGuage`

**Syntax**    `HCOPY:DEvice:COLor { OFF | ON | <NR1> }`  
`HCOPY:DEvice:COLor?`



**Arguments**    `OFF` or `<NR1> = 0` sets the hardcopy color mode to `OFF`.  
`ON` or `<NR1> ≠ 0` sets the hardcopy color mode to `ON`.  
At `*RST`, this value is set to `OFF`.

**Examples**    `HCOPY:DEvice:COLor ON`  
sets the hardcopy color mode to `ON`.

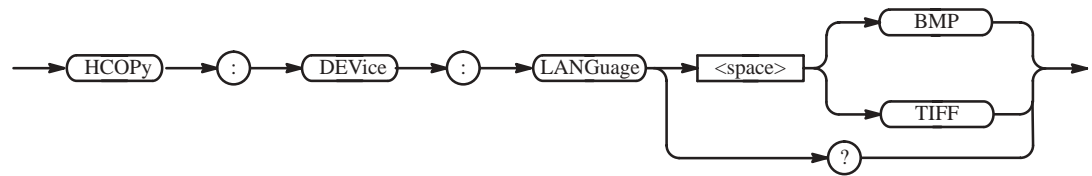
## HCOPY:DEvice:LANGuage (?)

This command sets the hardcopy data format.

**Group**    Hardcopy

**Related Commands**    `HCOPY:DEvice:COLor`

**Syntax** HCOpy:DEvice:LANGuage { BMP | TIFF }  
 HCOpy:DEvice:LANGuage?



**Arguments** BMP specifies the Windows bitmap file format.  
 TIFF specifies the TIFF format.  
 At \*RST, the parameter is set to BMP.

**Examples** HCOpy:DEvice:LANGuage TIFF  
 specifies the TIFF data format for hardcopy.

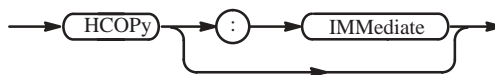
## HCOpy[:IMMediate] (No Query Form)

This command immediately initiates hardcopy output according to the current HCOpy setup parameters. For the waveform generator, this command is the same as HCOpy:SDUMp[:IMMediate]. For more information about hardcopy, see *Hardcopy Commands* on page 2–22.

**Group** Hardcopy

**Related Commands** HCOpy:DESTination, HCOpy:SDUMp[:IMMediate]

**Syntax** HCOpy[:IMMediate]



**Arguments** None

**Examples** HCOpy:IMMediate  
 starts hardcopy output.

## HCOPY:SDUMP[:IMMEDIATE] (No Query Form)

This command initiates a screen dump of the entire screen. For the waveform generator, this is the same as the HCOpy[:IMMEDIATE] command. For more information about hardcopy, see *Hardcopy Commands* on page 2–22.

**Group** Hardcopy

**Syntax** HCOpy:SDUMP[:IMMEDIATE]



**Arguments** None

**Examples** MMEORY:NAME "SAMPLE1.BMP";:HCOpy:SDUMP:IMMEDIATE  
prints the entire screen to the file SAMPLE1.BMP.

## \*IDN? (Query Only)

This command returns identification information for the waveform generator.

**Group** System

**Syntax** \*IDN?



**Arguments** None

**Returns** <manufacturer>, <model>, <serial\_number>, <firmware\_level>  
where  
 <manufacturer>::=SONY/TEK  
 <model>::={ AWG410 | AWG420 | AWG430 | AWG510 | AWG520 | AWG610 }  
 <serial\_number>::=0  
 <firmware\_level>::=SCPI:99.0 OS:x.y USR:x.y

**Examples** \*IDN?  
might return SONY/TEK,AWG410,0,SCPI:99.0 OS:1.0 USR:1.0

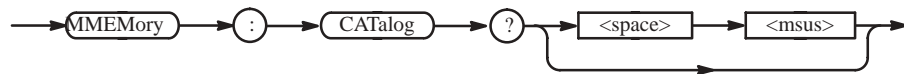
## MMEMemory:CATalog? (Query Only)

This command returns information about the current contents and state of the mass storage media.

**Group** Mass Memory

**Related Commands** MMEMemory:CDIRectory, MMEMemory:MSIS

**Syntax** MMEMemory:CATalog? [ <msus>]



**Arguments** <msus> (mass storage unit specifier)::=<string> is one of the following:

MAIN	The internal hard disk drive
FLOppy	The internal floppy disk drive
NET1, NET2, or NET3	The network drive 1, 2, or 3 (specified with the SYSTEM:COMMunicate:LAN commands)

**Returns** <NR1>,<NR1>[,<file\_name>,<file\_type>,<file\_size>]...

where:

The first <NR1> is the total amount of storage currently used, in bytes.  
For the network drives, <NR1> = 0.

The second <NR1> is the total amount of storage available.  
For the network drives, <NR1> = 0.

<file\_name>,<file\_type>,<file\_size>::=<string>

where

<file\_name> is the exact name of a file,

<file\_type> is DIR for directory, otherwise it is blank, and

<file\_size> is the size of the file, in bytes.

**Examples** MMEMemory:CATalog? "MAIN"  
might return the following response:  
484672,3878652,"SAMPLE1.WFM,2948"



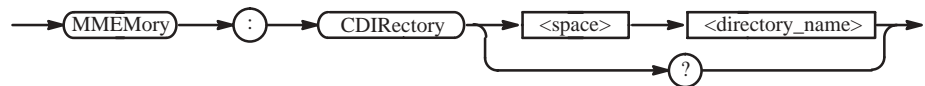
## MMEemory:CDIRectory (?)

This command changes the default directory for a mass memory file system. The default mass storage device is selected by MMEemory:MSIS command.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:MSIS

**Syntax** MMEemory:CDIRectory [<directory\_name>]  
MMEemory:CDIRectory?



**Arguments** <directory\_name>::=<string>  
is the default directory for a mass memory file system.

If you do not specify a parameter, the directory is set to the \*RST value.

At \*RST, this parameter is set to the root.

**Examples** MMEemory:CDIRectory "/AWG/WORK0"  
changes the default directory to /AWG/WORK0.

## MMEemory:CLOSe (No Query Form)

This command closes the file specified in the MMEemory:NAME command. This command is included only for compatibility with the SCPI standard and may not be used.

**Group** Mass Memory

**Related Commands** MMEemory:NAME, MMEemory:OPEN

**Syntax** MMEemory:CLOSe



**Arguments** None

**Examples** MMEemory:NAME "SAMPLE1.WFM";CLOSe  
closes the file SAMPLE1.WFM.

## MMEemory:COPY (No Query Form)

This command copies an existing file to a new file. An error is generated if the source file does not exist.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:DELeTe, MMEemory:MSIS

**Syntax** MMEemory:COPY <file\_source>,<file\_destination>



**Arguments** <file\_source>::=<file\_name>[,<msus>]  
<file\_destination>::=<file\_name>[,<msus>]

where:

<file\_name>::=<string> is the source or destination file name.

<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTem:COMMunicate:LAN commands)

**Examples** MMEemory:COPY "FILE1.WFM", "MAIN", "FILE2.WFM", "FLOppy"  
copies the file FILE1.WFM on the waveform generator hard disk to the file FILE2.WFM on the floppy disk.

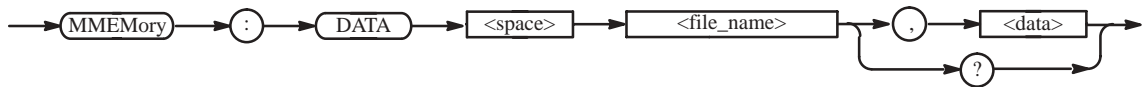
## MMEemory:DATA (?)

This command loads block data into the file on the default mass storage device, or returns the contents of the file.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:MSIS

**Syntax** MMEemory:DATA <file\_name>,<data>  
MMEemory:DATA?<file\_name>



**Arguments** <file\_name>::=<string> specifies the file to be loaded with data.  
<data> is in 488.2 block format.

**Examples** MMEemory:DATA "FILE1",#41024xxxxx...  
loads data into the file FILE1.

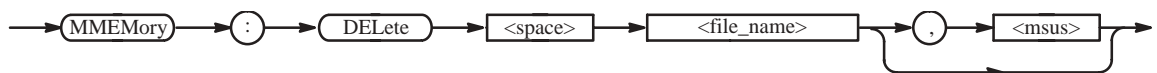
## MMEemory:DElete (No Query Form)

This command removes a file from the specified mass storage device.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:MSIS

**Syntax** MMEemory:DElete <file\_name>[,<msus>]



**Arguments** <file\_name>::=<string> specifies the file to be removed.

<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTem:COMMUnicate:LAN commands)

**Examples** MMEMory:DELeTe "FILE1.WFM", "FLOppy"  
removes the file FILE1.WFM on the floppy disk.

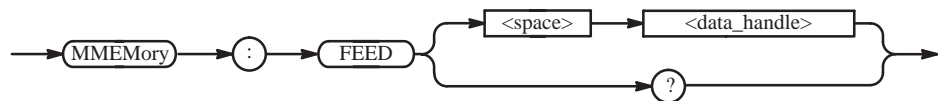
## MMEMory:FEED (?)

This command sets the data handle to be used to feed data into the file specified by MMEMory:NAME. For the waveform generator, the data handle is fixed to HCOPY. This command is included only for compatibility with the SCPI standard, and may not be used (refer to *Hardcopy Commands* on page 2–22).

**Group** Mass Memory

**Related Commands** MMEMory:NAME

**Syntax** MMEMory:FEED <data\_handle>  
MMEMory:FEED?



**Arguments** <data\_handle>::=<string> for the waveform generator, the data handle is fixed to HCOPY.

At \*RST, this parameter is set to "HCOP".

**Examples** MMEMory:FEED "HCOPY"  
sets the data handle.

## MMEMemory:INITialize (No Query Form)

This command initializes a specified mass storage media. In this application, you can initialize the internal hard disk or floppy disk.

---

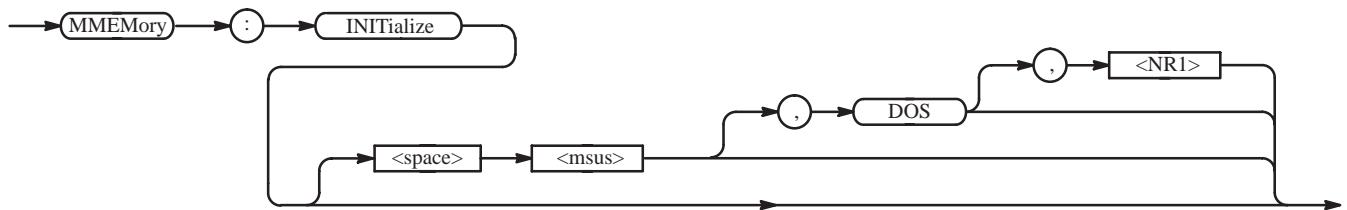
**NOTE.** *The initializing process erases all information that is already on the disk. This command is a "Quick Format" command, which cannot format MAC format, other OS format and damaged disk.*

---

**Group** Mass Memory

**Related Commands** MMEMemory:MSIS

**Syntax** MMEMemory:INITialize[ <msus>[,DOS[,<NR1>]]]



**Arguments** <msus> (mass storage unit specifier): :=<string> is the media on which the specified mass storage: { "MAIN" | "FLOppy" }

where MAIN means the internal hard disk, and FLOppy means the floppy disk.

The media is initialized in DOS format.

<NR1> is ignored in this application (It usually specifies media-dependent information.)

When you specify MAIN, this command returns the instrument settings to the factory defaults, except for the communication parameters (see *Appendix E: Factory Initialization Settings*).

**Examples** MMEMemory:INITialize "FLOppy"  
initializes a floppy disk in DOS format.

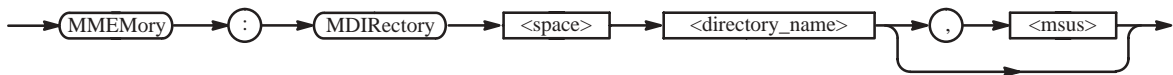
## MMEemory:MDIRectory (No Query Form)

This command creates a directory on the specified mass storage unit.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:MSIS

**Syntax** MMEemory:MDIRectory <directory\_name>[,<msus>]



**Arguments** <directory\_name>::=<string> specifies a new directory.

<msus> (mass storage unit specifier)::=<string> is the media on which you make the directory:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTEM:COMMUnicate:LAN commands)

**Examples** MMEemory:MDIRectory "WAVEFORM", "FLOppy"  
makes the directory "WAVEFORM" on the floppy disk.

## MMEemory:MOVE (No Query Form)

This command moves an existing file to another file name. If the source file does not exist, an error occurs.

**Group** Mass Memory

**Related Commands** MMEemory:CDIRectory, MMEemory:COPY, MMEemory:DELeTe, MMEemory:MSIS

**Syntax** MMEemory:MOVE <file\_source>,<file\_destination>



**Arguments** <file\_source>, <file\_destination>  
 ::=<file\_name>[,<msus>]

where:

<file\_name>::=<string> is the source or destination file name.

<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTEM:COMMunicate:LAN commands)

**Examples** MMEemory:MOVE "FILE1.WFM", "MAIN", "FILE2.WFM", "FLOppy"  
 moves the file FILE1.WMF on the waveform generator hard disk to FILE2.WFM on the floppy disk.

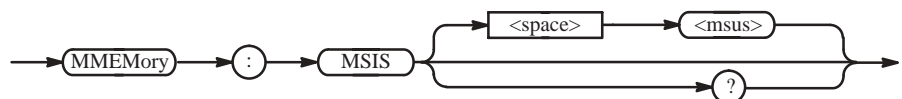
## MMEemory:MSIS (?)

The “Mass Storage IS” command selects a default mass storage device that is used by all MMEemory commands except INITialize.

**Group** Mass Memory

**Related Commands** All MMEemory commands except INITialize.

**Syntax** MMEemory:MSIS[ <msus>]  
 MMEemory:MSIS?



**Arguments** <msus>(Mass Storage Unit Specifier)::=<string> specifies a default mass storage device.

MAIN	Internal hard disk drive
FLOppy	Internal floppy disk drive
NET1, NET2, or NET3	Network drive 1, 2, or 3 (specified with the SYSTEM:COMMunicate:LAN commands)

At \*RST, this parameter is set to MAIN.

**Examples**    MMEMory:MSIS "FLOppy"  
 selects the floppy disk drive as the default mass storage device.

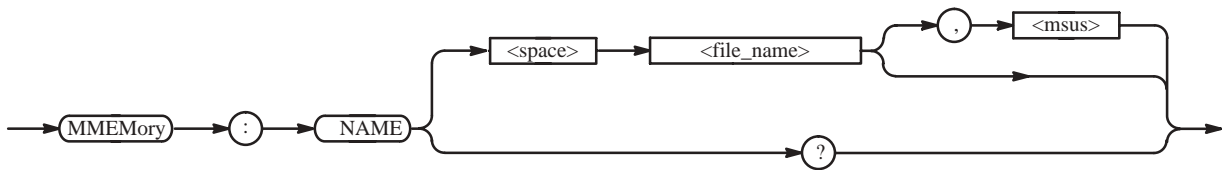
## MMEMory:NAME (?)

This command specifies the name of the file specification used by MMEMo-ry:OPEN or CLOSe commands.

**Group**        Mass Memory

**Related Commands**    MMEMory:CLOSe, MMEMory:OPEN

**Syntax**        MMEMory:NAME <file\_name>[,<msus>]  
 MMEMory:NAME?



**Arguments**    <file\_name>::=<string> is the name of the file to be opened or closed.

<msus> (mass storage unit specifier)::=<string> is the media on which the file exists:

- |                     |   |
|---------------------|---|
| MAIN                | Internal hard disk drive  |
| FLOppy              | Internal floppy disk drive  |
| NET1, NET2, or NET3 | Network drive 1, 2, or 3 (specified with the SYSTem:COMMUnicate:LAN commands) |

At \*RST, this parameter is set to "HARDCOPY".

**Examples**    MMEMory:NAME "SAMPLE1.WFM", "NET1";OPEN  
 opens the file SAMPLE1.WFM on the network drive 1.



## MMEMory:OPEN (No Query Form)

This command opens the file specified in the MMEMory:NAME command. This command is included only for compatibility, and may not be used.

**Group** Mass Memory

**Related Commands** MMEMory:CDIRectory, MMEMory:CLOSe, MMEMory:MSIS, MMEMory:NAME

**Syntax** MMEMory:OPEN



**Arguments** None

**Examples** MMEMory:NAME "SAMPLE1.WFM", "NET1";OPEN  
opens the file SAMPLE1.WFM on the network drive 1.

## \*OPC (?)

Operation complete command (query). Use this command between two other commands to ensure completion of the first command before processing the second command.

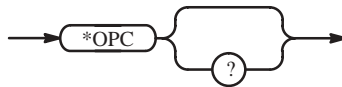
In this application, all commands are designed to be executed in the order in which they are sent from the external controller. The \*OPC (?) command is included to ensure compliance with the SCPI standard. You do not need to use this command.

Refer to page 3–6 about the OPC bit of SESR (Standard Event Status Register).

**Group** Synchronization

**Related Commands** \*WAI

**Syntax** \*OPC  
\*OPC?



**Arguments** None

**Returns** <NR1>=1 when all pending operations are finished.

## \*OPT? (Query Only)

This command returns the implemented options of the waveform generator.

**Group** System

**Syntax** \*OPT?



**Arguments** None

**Returns** <opt>[,<opt>[,<opt>[,<opt>]]]

where:

- 0 the waveform generator has no options installed.
- 01 the waveform generator has Long Memory Option installed.  
(AWG400 series only)
- 03 the waveform generator has Digital Data Out Option installed.  
(AWG400 series only)
- 05 the waveform generator has Single-end Output Option installed.  
(AWG400 series only)
- 10 the waveform generator has ATE Option installed.
- DD0 the waveform generator has Option 03 (Digital Data Out) installed.  
(AWG500 series only)

**Examples** \*OPT?  
might return 0 to indicate that no option is installed in the instrument.

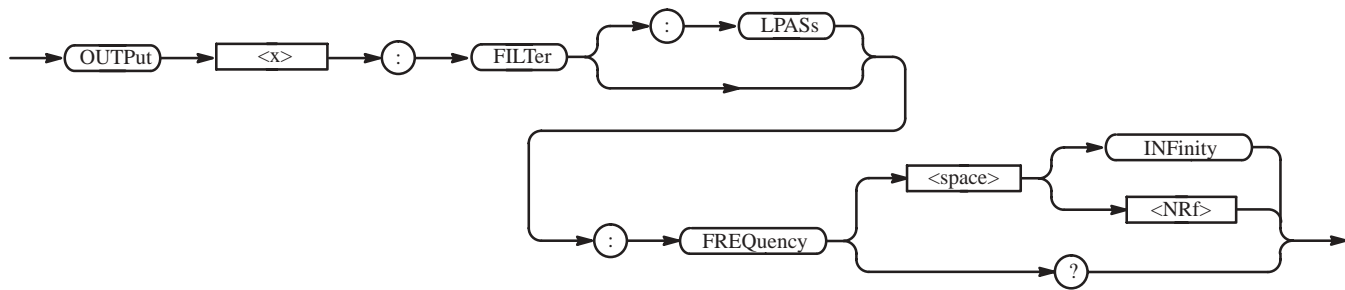
## OUTPut<x>:FILTer[:LPASs]:FREQuency (?)

This command determines the cutoff frequency of the low pass filter for a specified channel.

**Group** Output

**Syntax** OUTPut<x>:FILTer[:LPASs]:FREQuency { <NRf> | INFinity }  
OUTPut<x>:FILTer[:LPASs]:FREQuency?  
<x> ::= following

AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
1	1, 2	1, 2, 3	1	1, 2	1



**Arguments**

<NRf> is the cutoff frequency of the low pass filter, in Hz.  
 In AWG400 series, you can select 1e6 (1MHz), 5e6 (5MHz), 20e6 (20MHz), 50e6 (50MHz), or 9.9e37 (INFinity, that means “through”).  
 In AWG500 series, you can select 10e6 (10MHz), 20e6 (20MHz), 50e6 (50MHz), 100e6 (100MHz), or 9.9e37 (INFinity, that means “through”).  
 In AWG600 series, you can select 20e6 (20MHz), 50e6 (50MHz), 100e6 (100MHz), 200e6 (200MHz), or 9.9e37 (INFinity, that means “through”).  
 At \*RST, this value is set to 9.9e37 (“through”).

**Examples**

OUTPut1:FILTer:LPASs:FREQuency 100e6  
 sets the cutoff frequency of the low pass filter for CH 1 to 100 MHz.  
 At \*RST, this value is set to 9.9e37 (“through”).

**OUTPut<x>:ISTate (?)**

**AWG400 series(except Op05) , AWG510 and AWG610 Only**

Controls whether the waveform generator inverted CH 1 output terminal ( $\overline{CH1}$ ) is enabled or disabled. When the function is OFF, the CH1 terminal is at maximum isolation from the signal.

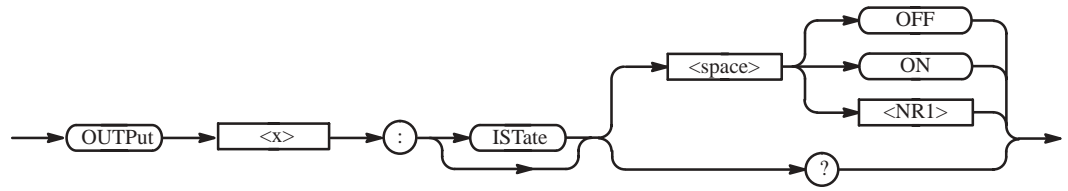
**Group**

Output

**Syntax**

OUTPut<x>:ISTate { ON | OFF | <NR1> }  
 OUTPut<x>:ISTate?  
 <x> ::= following

except AWG410 Op05	except AWG420 Op05	except AWG430 Op05	AWG510	AWG610
1	1, 2	1, 2, 3	1	1



**Arguments** <ON> or <NR1> ≠ 0 enables the  $\overline{\text{CHI}}$  output.  
 <OFF> or <NR1> = 0 disables the  $\overline{\text{CHI}}$  output.  
 At \*RST, this value is set to 0 (OFF).

**Examples** OUTPUT1:ISTate ON  
 enables the  $\overline{\text{CHI}}$  output.

## OUTPut<x>[:STATe] (?)

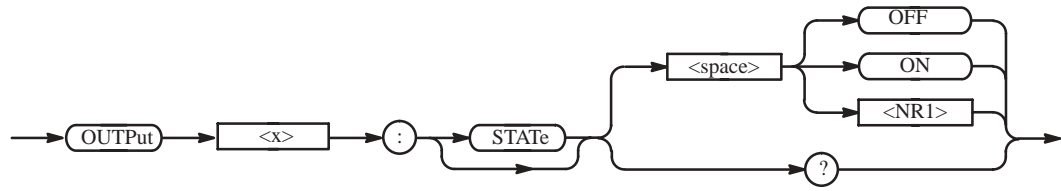
This command controls whether the output terminal is open or closed. When the function is OFF, the terminal is at maximum isolation from the signal.

**Group** Output

**Related Commands** SOURce1:COMBine:FEED

**Syntax** OUTPut<x>[:STATe] { ON | OFF | <NR1> }  
 OUTPut<x>[:STATe]?  
 <x> ::= following

AWG410	AWG410 Op03	AWG420	AWG420 Op03	AWG430	AWG430 Op03	AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 4	1, 2	1,2,4,5	1,2,3	1,2,3,4,5 ,6	1, 7	1,5,7	1,2,7	1,2,5,7	1



**Arguments** <ON> or <NR1> ≠ 0 turns the output on.

<OFF> or <NR1> = 0 turns the output off.

OUTPut7[:STATe] can not be set to ON when SOURce1:COMBine:FEED is specified as SOURce7 (the noise generator).

At \*RST, this value is set to 0 (OFF).

**Examples** OUTPut1:STATe ON  
turns the CH 1 output on.

## \*PSC (?)

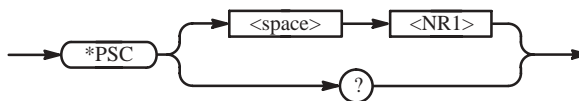
This command sets and queries the power-on status flag that controls the automatic power-on handling of the SRER, ESER, OENR, and QENR registers. When \*PSC is true, the registers are set to 0 at power-on. When \*PSC is false, the current values in the registers are preserved in nonvolatile memory when power is shut off, and then are restored at power-on. For a complete discussion of the use of these registers, refer to the *Status and Event Reporting* section on page 3-1.

**Group** Status

**Related Commands** \*ESE, \*SRE, STATus:OPERation:ENABle, STATus:QUESTionable:ENABle

**Syntax** \*PSC <NR1>

\*PSC?



**Arguments** <NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear, and allows the waveform generator to possibly assert SRQ after power-on.

<NR1> ≠ 0 sets the power-on status clear flag true. Sending \*PSC 1 therefore enables the power-on status clear and prevents any SRQ assertion after power-on. Using an out-of-range value causes an execution error.

**Examples** \*PSC 0  
sets the power-on status clear flag to false.

\*PSC?  
might return the value 1, showing that the power-on status clear flag is set to true.

## \*RST (No Query Form)

This command resets the waveform generator to the default state. This command has no effect on the network and communication settings, such as GPIB or IP address. Refer to *Appendix E: Factory Initialization Settings*.

**Group** System

**Related Commands** SYSTem:SECurity:IMMEDIATE

**Syntax** \*RST



**Arguments** None

**Examples** \*RST  
resets the instrument.

## [SOURce<x>]:COMBine:FEED (?)

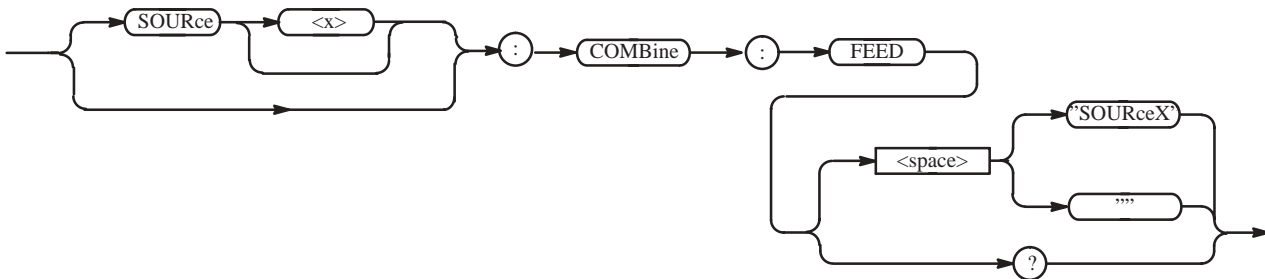
*AWG400/500 series Only*

Adds the signal from the noise generator or the external input to the CH 1 output, or opens these addition lines to the CH 1 output.

**Group** Source

**Related Commands** OUTPut7:STATe

**Syntax** [SOURce1]:COMBine:FEED { "SOURce7" | "SOURce8" | "" }  
 (AWG400/500 series)  
 SOURce2:COMBine:FEED { "SOURce9" | "SOURce10" | "" }  
 (AWG420/430)  
 SOURce3:COMBine:FEED { "SOURce11" | "SOURce12" | "" }  
 (AWG430)  
 [SOURce[1|2|3]:]COMBine:FEED?



**Arguments**

SOURce1: SOURce7 SOURce8 ""(null)	adds the signal from the noise generator to the CH 1 output. adds the signal from the external input to the CH 1 output. opens the noise generator output line or the external input line to the CH 1 output, i.e. adds no signal.
SOURce2: SOURce9 SOURce10 ""(null)	adds the signal from the noise generator to the CH 2 output. adds the signal from the external input to the CH 2 output. opens the noise generator output line or the external input line to the CH 2 output, i.e. adds no signal.
SOURce3: SOURce11 SOURce12 ""(null)	adds the signal from the noise generator to the CH 3 output. adds the signal from the external input to the CH 3 output. opens the noise generator output line or the external input line to the CH 3 output, i.e. adds no signal.

At \*RST, the parameter is set to "" (null).

**Examples**

SOURce1:COMBine:FEED "SOURce7"  
 adds noise to the CH 1 output. Then,

SOURce1:COMBine:FEED ""  
 opens the noise generator output line to the CH 1 output.



## [SOURce<x>]:FREQuency[:CW|FIXed] (?)

This command sets the sampling frequency to output a waveform or pattern file. The file is specified by the SOURce<x>:FUNction:USER command.

CW (Continuous Wave) and FIXed are aliases, and have the same effect.

**Group** Source

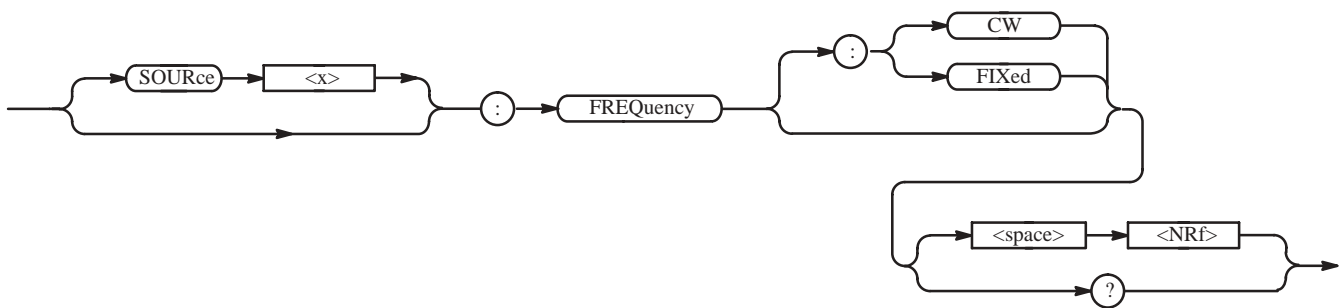
**Related Commands** [SOURce<x>]:FUNction:USER

**Syntax** [SOURce<x>]:FREQuency[:CW|FIXed] <NRf>

[SOURce<x>]:FREQuency[:CW|FIXed]?

<x> ::= following

AWG410	AWG410 Op03	AWG420	AWG420 Op03	AWG430	AWG430 Op03	AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 4	1, 2	1,2,4,5	1,2,3	1,2,3,4,5 ,6	1	1, 5	1, 2	1,2,5	1



**Arguments** In a case of AWG, SOURce1 to SOURce6 are equivalent.

<NRf> is the sampling frequency. The range is as follows.

AWG400 series : 10 kHz to 200 MHz.

AWG500 series : 50 kHz to 1 GHz.

AWG600 series : 50 kHz to 2.6 GHz.

At \*RST, this value is set to 100 MHz.

**Examples**     SOURce1:FREQuency 10MHz  
                   sets the sampling frequency to 10 MHz.

## [SOURce<x>]:FUNction:USER (?)

This command specifies a waveform or pattern file that you created as the output source. This command loads the file into the waveform generator's RAM prior to output.

**Group**        Source

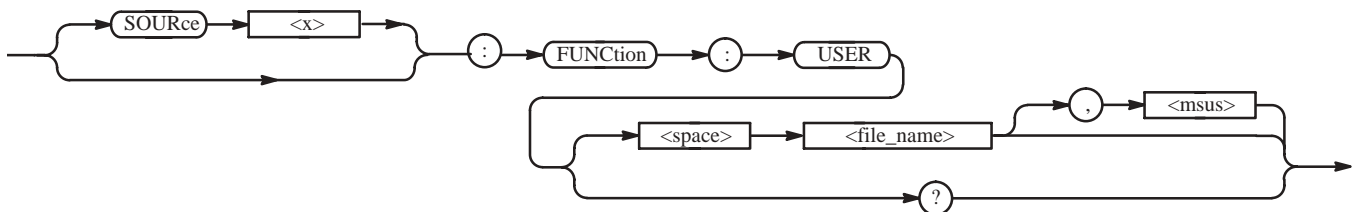
**Related Commands**     [SOURce<x>]:FREQuency[:CW|FIXed]

**Syntax**        [SOURce<x>]:FUNction:USER <file\_name>[,<msus>]

[SOURce<x>]:FUNction:USER?

<x> ::= following

AWG410	AWG410 Op03	AWG420	AWG420 Op03	AWG430	AWG430 Op03	AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 4	1, 2	1,2,4,5	1,2,3	1,2,3,4,5 ,6	1	1, 5	1, 2	1,2,5	1



**Arguments**     In AWG400 series, SOURce1 and SOURce4, SOURce2 and SOURce5, SOURce3 and SOURce6, are equivalent.  
                   In AWG520Op03, SOURce2 and SOURce5 are same as each the other.

<file\_name> ::= <string> is the name of a waveform or pattern file to output.

<msus> (mass storage unit specifier) ::= <string> is the media on which the file exists:

MAIN                             The internal hard disk drive  
 FLOppy                          The internal floppy disk drive

NET1, NET2, or NET3 The network drive 1, 2, or 3 (specified with the SYSTem:COMMunicate:LAN commands)

At \*RST, this value is set to "" (null).

**Examples** SOURce1:FUNction:USER "SAMPLE1.WFM", "FLOppy"  
 specifies the file SAMPLE1.WFM on the floppy disk as the CH 1 output source.

## [SOURce<x>]:MARKer[1|2]:DELay (?)

*AWG500/600 series Only*

This command sets the marker output delay referenced to the continuous clock output for a specified channel.

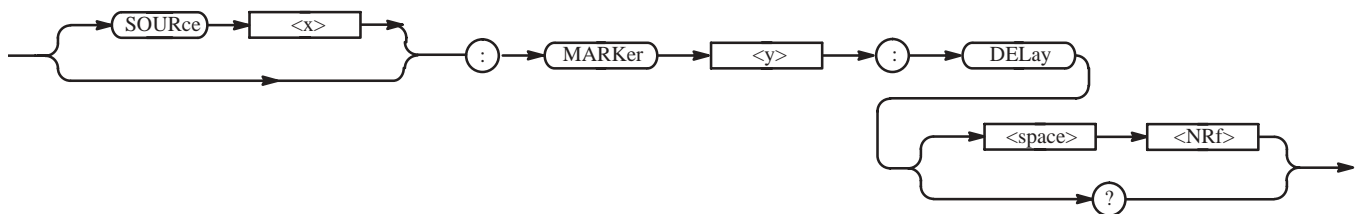
**Group** Source

**Syntax** [SOURce<x>]:MARKer[1|2]:DELay <NRf>

[SOURce<x>]:MARKer[1|2]:DELay?

<x> ::= following

AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 5	1, 2	1, 2, 5	1



**Arguments** <NRf> is the delay time in seconds. The range is as follows:

AWG500 series : 0 ns to +2 ns with a resolution of 20 ps.  
 AWG600 series : 0 ns to +1.5 ns with a resolution of 100 ps.

At \*RST, this value is set to 0.

**Examples**    `SOURce1:MARKer1:DELay 500ps`  
 sets the delay of marker1 to 500 ps for CH 1 output.

## **[SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:HIGH (?)**

*AWG500/600 series Only*

This command sets the high level for the marker output.

**Group**        Source

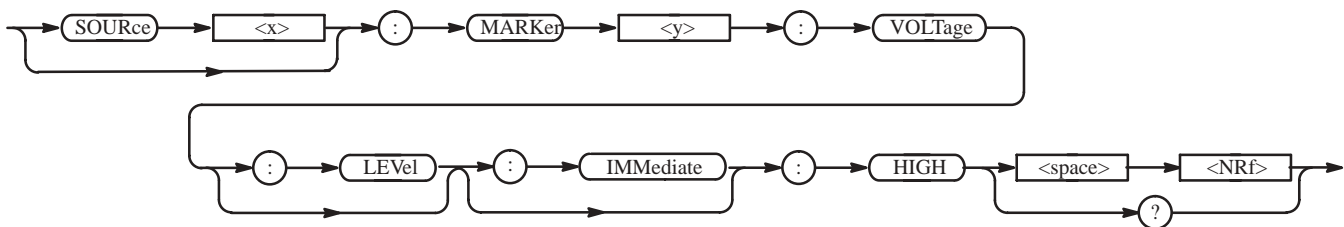
**Related Commands**    `[SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:LOW`

**Syntax**        `[SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:HIGH <Nrf>`

`[SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:HIGH?`

`<x> ::= following`

AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 5	1, 2	1, 2, 5	1



**Arguments**    `<Nrf>` is the high level voltage of the marker output. Note that the high level must be larger than the low level. The range is as follows:

AWG500 series : -2.0 V to 2.0 V (into 50 Ω) with a resolution of 0.05 V.

AWG600 series : -1.1 V to 3.0 V (into 50 Ω) with a resolution of 0.05 V. Note that the difference between high and low level is restricted to within 2.5 V.

At \*RST, this value is set to 2 V.

**Examples**        `SOURce1:MARKer1:VOLTage:LEVel:IMMediate:HIGH 1.2`  
 sets the high level of the marker 1 output on CH 1 to 1.2 V.

**[SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:LOW (?)****AWG500/600 series Only**

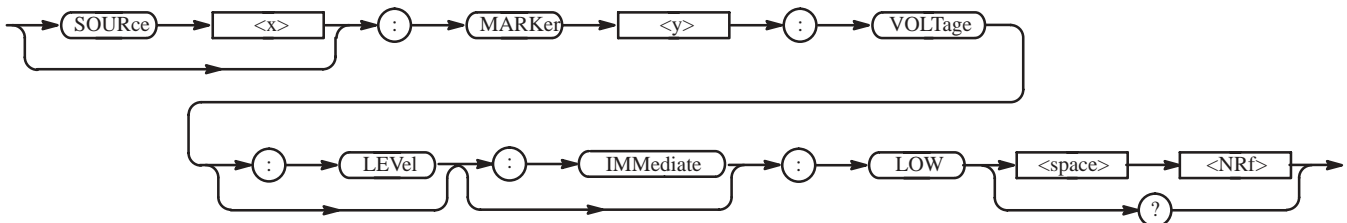
This command sets the low level voltage for the marker output.

**Group** Source**Related Commands** [SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:HIGH**Syntax** [SOURce<x>]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:LOW <NRf>

[SOURce&lt;x&gt;]:MARKer[1|2]:VOLTage[:LEVel][:IMMediate]:LOW?

&lt;x&gt; ::= following

AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 5	1, 2	1, 2, 5	1

**Arguments** <NRf> is the low level voltage of the marker output. Note that the high level must be larger than the low level. The range is as follows:

AWG500 series : -2.0 V to 2.0 V (into 50 Ω) with a resolution of 0.05 V.

AWG600 series : -1.1 V to 3.0 V (into 50 Ω) with a resolution of 0.05 V. Note that the difference between high and low level is restricted to within 2.5 V.

At \*RST, this value is set to 0.

**Examples** SOURce1:MARKer1:VOLTage:LEVel:IMMediate:LOW -1.2  
sets the low level voltage of the marker 1 output on CH 1 to -1.2 V.

## SOURce<x>:POWer[:LEVel][:IMMediate][:AMPLitude] (?)

**AWG400/500 series Only**

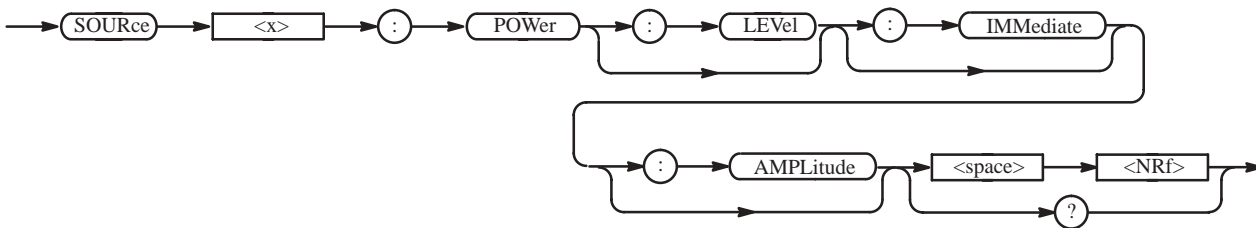
Sets the level of the noise generator output, in dBm/Hz.

**Group** Source

**Syntax** SOURce<x>:POWer[:LEVel][:IMMediate][:AMPLitude] <NRf>  
 SOURce<x>:POWer[:LEVel][:IMMediate][:AMPLitude]?

<x> ::= following

AWG410	AWG420	AWG430	AWG510	AWG520
7	7, 9	7, 9, 11	7	7



**Arguments** <NRf> is the noise output level.

AWG400 series :

The range is -140 to -95 dBm/Hz in 1 dB step.

The range is -130 to -95 dBm/Hz in 1 dB step.(opt05, Single act)

AWG500 series :

The range is -145 to -105 dBm/Hz in 1 dB step.

At \*RST, this value is set to -95 dBm/Hz(AWG400 series) or -105 dBm/Hz (AWG500 series).

**Examples** SOURce7:POWer:LEVel:IMMediate:AMPLitude -120  
 sets the level of the noise generator output to -120 dBm/Hz.

## [SOURce<x>]:ROSCillator:SOURce (?)

This command selects the reference oscillator.

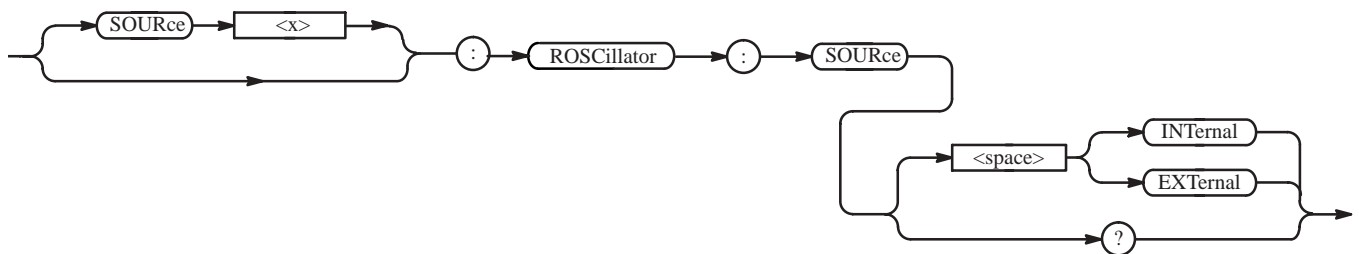
**Group** Source

**Syntax** [SOURce<x>]:ROSCillator:SOURce { INTernal | EXTernal }

[SOURce<x>]:ROSCillator:SOURce?

<x> ::= following

AWG410	AWG410 Op03	AWG420	AWG420 Op03	AWG430	AWG430 Op03	AWG510	AWG510 Op03	AWG520	AWG520 Op03	AWG610
1	1, 4	1, 2	1,2,4,5	1,2,3	1,2,3,4,5 ,6	1	1, 5	1, 2	1,2,5	1



**Arguments** Suffix 1 through 6 are same.

INTernal means that the reference frequency is derived from the internal precision oscillator.

EXTernal means the reference frequency is derived from an external signal supplied through the Reference Clock Input connector.

At \*RST, this parameter is set to INTernal.

**Examples** SOURce1:ROSCillator:SOURce EXTernal  
selects the external clock source.

## SOURce[2|3|5|6]:SKEW (?)

*AWG400 series Only*

This command sets the skew.

**Group** Source

**Syntax** SOURce[2|3|5|6]:SKEW <NRf>  
SOURce[2|3|5|6]:SKEW?



**Arguments** <NRf> is the skew:  
The range is  $-2.52\text{ns}$  to  $2.52\text{ns}$ , in 70ps steps.

**Examples** SOURce2:SKEW 2.52ns  
sets the skew of CH 2 to 2.52ns.

## [SOURce<x>]:VOLTage[:LEVel][:IMMediate][:AMPLitude] (?)

This command sets the actual magnitude of the output signal.

**Group** Source

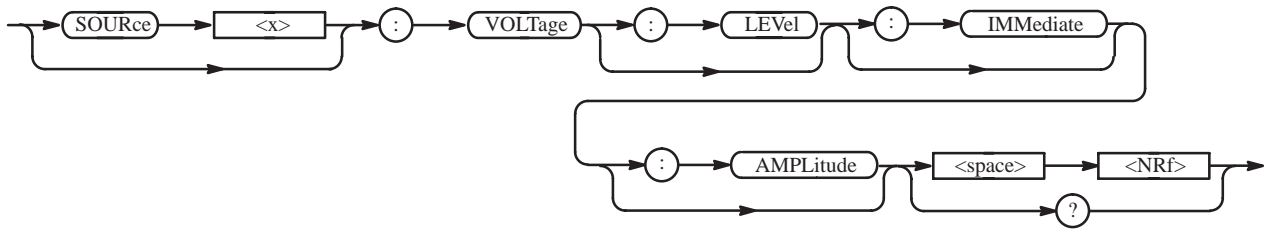
**Related Commands** [SOURce<x>]:VOLTage[:LEVel][:IMMediate]:OFFSet

**Syntax** [SOURce<x>]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf>  
[SOURce<x>]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?

<x> ::= following

AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
1	1, 2	1, 2, 3	1	1, 2	1



**Arguments**

<NRf> is the amplitude:

Range(AWG500/600 series)

20 mV to 2.0 V (into 50  $\Omega$ ), in 1 mV steps.

In AWG600 series, note that when DOUT is set to 1 (ON), the range is 20 mV to 1.0 V.

Range(AWG400 series, Differential, Normal)

20 mV to 2.0 V (into 50  $\Omega$ ), in 1 mV steps.

Range(AWG400 series, Differential, Low dist.)

20 mV to 0.5 V (into 50  $\Omega$ ), in 1 mV steps.

Range(AWG400 series opt05, Single act, Normal)

20 mV to 5.0 V (into 50  $\Omega$ ), in 1 mV steps.

Range(AWG400 series opt05, Single act, Low dist.)

20 mV to 0.5 V (into 50  $\Omega$ ), in 1 mV steps.

At \*RST, this value is set to 1 V.

**Examples**

SOURce1:VOLTage:LEVel:IMMediate:AMPLitude 230mV  
sets the amplitude of CH 1 waveform to 230 mV.

**SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH (?)****AWG500 series Only**

This command sets the high level of the signal from the pattern generator (SOURce5, Option 03). This command is used in conjunction with SOURce5:VOLTage[:LEVel][:IMMediate]:LOW.

**Group**

Source

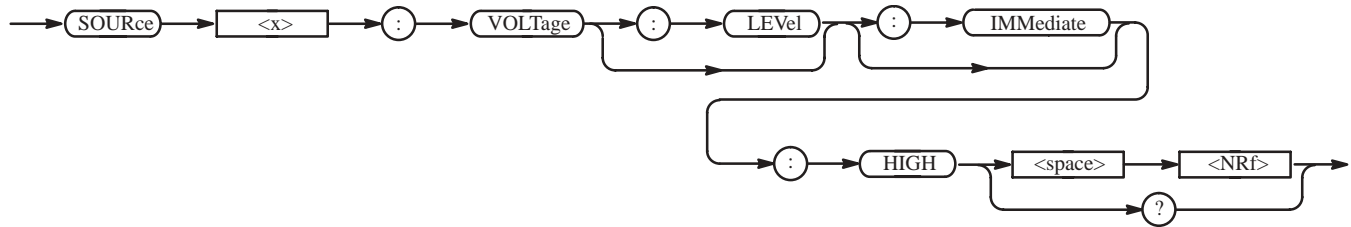
**Related Commands**

SOURce5:VOLTage[:LEVel][:IMMediate]:LOW

**Syntax**

SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH <NRf>

SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH?



**Arguments** <NRf> is the high level of the pattern generator output. The range is -2.0 V to +2.0 V in 1mV steps.

At \*RST, this value is set to 2 V.

**Examples** SOURce5:VOLTage:LEVel:IMMediate:HIGH 230mV sets the high level of the pattern generator output to 230 mV.

## SOURce5:VOLTage[:LEVel][:IMMediate]:LOW (?)

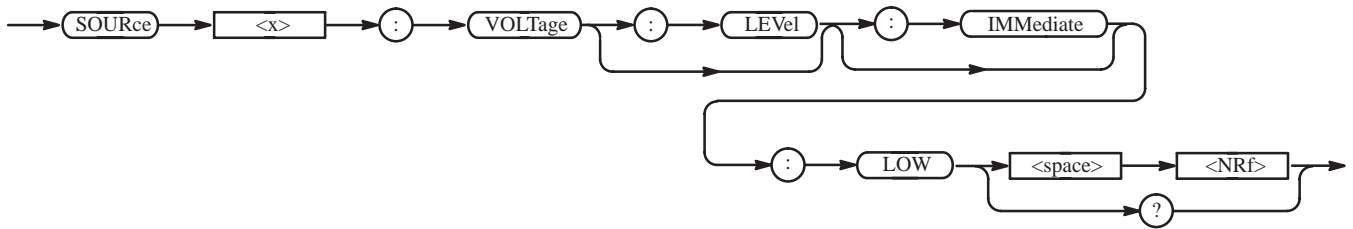
*AWG500 series Only*

This command sets the low level of the signal from the pattern generator (SOURce5, Option 03). This command is used in conjunction with SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH.

**Group** Source

**Related Commands** SOURce5:VOLTage[:LEVel][:IMMediate]:HIGH

**Syntax** SOURce5:VOLTage[:LEVel][:IMMediate]:LOW <NRf>  
SOURce5:VOLTage[:LEVel][:IMMediate]:LOW?



**Arguments** <NRf> is the low level of the pattern generator output in volts. The range is -2.0 V to +2.0 V in 1 mV steps.  
At \*RST, this value is set to 0.

**Examples** SOURCE5:VOLTage:LEVel:IMMediate:LOW -230mV sets the low level of the pattern generator output to -230 mV.

### [SOURCE<x>]:VOLTage[:LEVel[:IMMediate]:OFFSet (?)

This command sets the non-time-varying component of the signal that is added to SOURCE1 (CH 1), SOURCE2 (CH 2), or SOURCE3 (CH 3).

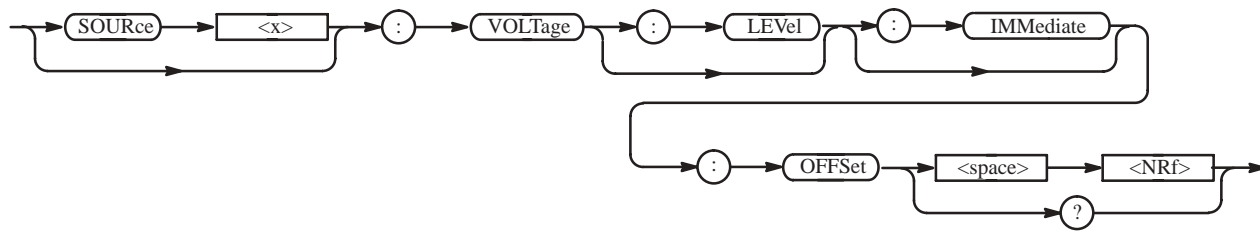
**Group** Source

**Related Commands** [SOURCE<x>]:VOLTage[:LEVel[:IMMediate][:AMPLitude]

**Syntax** [SOURCE<x>]:VOLTage[:LEVel[:IMMediate]:OFFSet <NRf>  
[SOURCE<x>]:VOLTage[:LEVel[:IMMediate]:OFFSet?

<x> ::= following

AWG410	AWG420	AWG430	AWG510	AWG520	AWG610
1	1, 2	1, 2, 3	1	1, 2	1



**Arguments**

<NRf> is the offset voltage.  
 The range(AWG500/600 series) is -1.000 V to +1.000 V, in 1 mV steps.  
 The range(AWG400 series, Complementary, Normal) is -1.000 V to +1.000 V, in 1 mV steps.  
 The range(AWG400 series, Complementary, Low Dist.) is No offset.  
 The range(AWG400 series opt05, Single end, Normal) is -2.500 V to +2.500 V, in 1 mV steps.  
 The range(AWG400 series opt05, Single end, Low Dist.) is No offset.  
 At \*RST, this value is set to 0.

**Examples**

SOURce1:VOLTage:LEVel:IMMEDIATE:OFFSet 50mV  
 sets the offset voltage of the CH 1 output to 50 mV.

**\*SRE (?)**

This command sets and queries the bits in the Service Request Enable Register (SRER). For a complete discussion of the use of these registers, refer to the *Status and Events* section of this manual.

**Group**

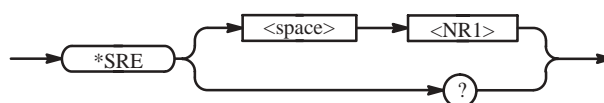
Status

**Related Commands**

\*CLS, \*ESE, \*ESR?, \*PSC, \*STB?

**Syntax**

\*SRE <NR1>  
 \*SRE?



- Arguments** <NR1> is a value in the range from 0 to 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if \*PSC is 1. If \*PSC is 0, the SRER maintains its value through a power cycle.
- Examples** \*SRE 48  
sets the bits in the SRER to the binary value 00110000.
- \*SRE?  
might return a value of 32, showing that the bits in the SRER have the binary value 00100000.

## STATus:OPERation:CONDition? (Query Only)

This command returns the contents of the Operation Condition Register (OCR). For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** STATus:OPERation:ENABle, STATus:OPERation[:EVENT]?

**Syntax** STATus:OPERation:CONDition?



**Arguments** None

**Returns** <NR1> indicates that the content of the OCR in a decimal number.

**Examples** STATus:OPERation:CONDition?  
might return 32, which indicates that the OCR contains the binary number 00000000 00100000, and the instrument is waiting for trigger.

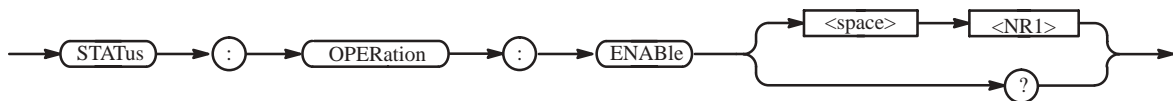
## STATus:OPERation:ENABLE (?)

This command sets the enable mask for the Operation Enable Register (OENR). For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** STATus:OPERation:CONDition?, STATus:OPERation[:EVENT]?

**Syntax** STATus:OPERation:ENABLE <NR1>  
 STATus:OPERation:ENABLE?



**Arguments** <NR1> is the enable mask for the OENR. The range is 0 to 65535.

**Returns** <NR1> indicates that the content of the OENR in a decimal number.

**Examples** STATus:OPERation:ENABLE 1  
 sets the CALibrating bit in the OENR to “enable”.

STATus:OPERation:ENABLE?  
 might return 1 which indicates that the OENR contains the binary number 00000000 00000001, and the CAL bit is set to “enable”.

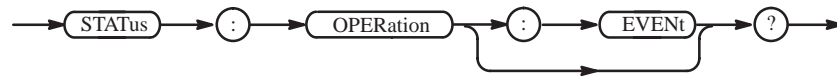
## STATus:OPERation[:EVENT]? (Query Only)

This command returns the contents of the Operation Event Register (OEVr) and clears it. For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** STATus:OPERation:CONDition?, STATus:OPERation:ENABLE

**Syntax** STATus:OPERation[:EVENT]?



**Returns** <NR1> indicates the content of the OEVR in a decimal number.

**Examples** STATus:OPERation:EVENT?  
might return 1, which indicates that the OEVR contains the binary number 00000000 00000001, and the CAL bit is set.

## STATus:PRESet (No Query Form)

This command presets the SCPI enable registers OENR and QENR. For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Syntax** STATus:PRESet



**Arguments** None

**Examples** STATus:PRESet  
presets the SCPI enable registers.

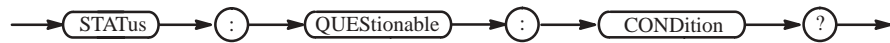
## STATus:QUESTionable:CONDition? (Query Only)

This command returns the contents of the Questionable Condition Register (QCR). For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** STATus:QUESTionable:ENABLE, STATus:QUESTionable[:EVENT]?

**Syntax** STATus:QUESTionable:CONDition?



**Returns** <NR1> indicates that the content of the QCR in a decimal number.

**Examples** `STATUS:QUESTIONABLE:CONDITION?`  
 might return 32, which indicates that the QCR contains the binary number 00000000 00100000, and the accuracy of frequency is questionable.

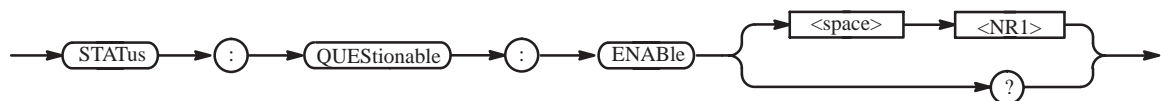
## STATUS:QUESTIONABLE:ENABLE (?)

This command sets the enable mask for the Questionable Enable Register (QENR). For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** `STATUS:QUESTIONABLE:CONDITION?`, `STATUS:QUESTIONABLE[:EVENT]?`

**Syntax** `STATUS:QUESTIONABLE:ENABLE <NR1>`  
`STATUS:QUESTIONABLE:ENABLE?`



**Arguments** <NR1> is the content of the QENR. The range is 0 to 65535.

**Returns** <NR1> indicates that the content of the QENR in a decimal number.

**Examples** `STATUS:QUESTIONABLE:ENABLE #H20`  
 sets the FREQUENCY bit in the QENR to “enable”.

`STATUS:QUESTIONABLE:ENABLE?`  
 might return 32, which indicates that the QENR contains the binary number 00000000 00100000, and the FREQ bit is set to “enable.”



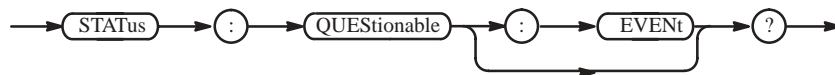
## STATus:QUESTionable[:EVENT]? (Query Only)

This command returns the contents of the Questionable Event Register (QEVr) and clears it. For more information on registers, refer to the *Status and Events* section of this manual.

**Group** Status

**Related Commands** STATus:QUESTionable:CONDition?, STATus:QUESTionable:ENABle

**Syntax** STATus:QUESTionable[:EVENT]?



**Returns** <NR1> indicates that the contents of the QEVr in a decimal number.

**Examples** STATus:QUESTionable:EVENT?  
might return 32, which indicates that the QEVr contains the binary number 00000000 00100000, and the FREQ bit is set”.

## STATus:QUEue[:NEXT]? (Query Only)

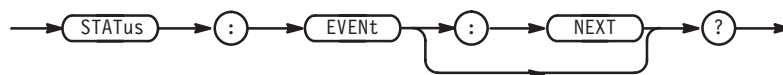
*AWG500/600 series Only*

This command returns the next item from the error/event queue and removes that item from the queue. Operation is identical to that of the SYSTem:ERRor? query. Refer to the *Status and Events* section of this manual for more details.

**Group** Status

**Related Commands** SYSTem:ERRor?

**Syntax** STATus:QUEue[:NEXT]?



**Returns** <Error/event number>,"<Error/event description>  
[;<Device dependent info>]"

where

<Error/event number> is an integer between -32768 and 32767.

0 indicates that no error or event has occurred.

Positive values are error/event numbers determined by this instrument.

Negative values are error/event numbers reserved in SCPI standards.

<Error/event description> is a message relating to the error/event number.

<Device dependent info> is more detailed information relating to the error/event number.

**Examples**

STATus:QUEue:NEXT?

might return the following response:

-102,"Syntax error;possible invalid suffix - :SOUR:FREQ 2V"

In this case, the unit is invalid.

**\*STB? (Query Only)**

This command returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For a complete discussion of the use of these registers, refer to the *Status and Events* section of this manual.

**Group**

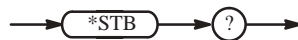
Status

**Related Commands**

\*CLS, \*ESE, \*ESR?, \*SRE

**Syntax**

\*STB?



**Arguments**

None

**Returns**

<NR1> indicates that the content of the SBR in a decimal number.

**Examples**

\*STB?

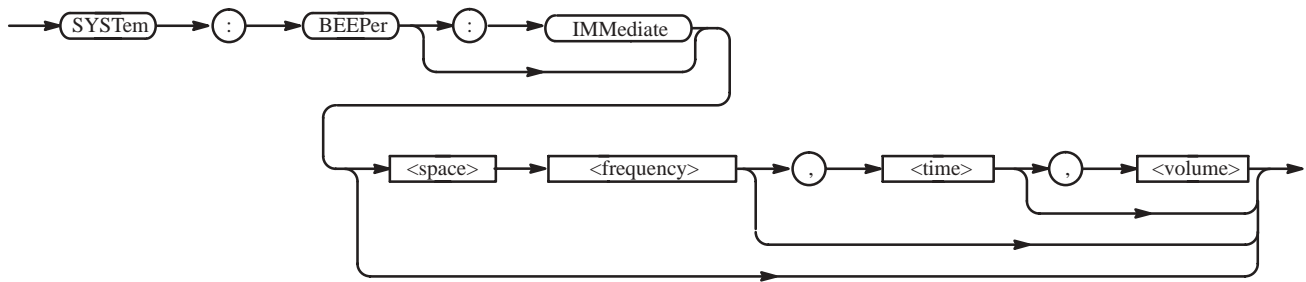
might return 96, which indicates that the SBR contains the binary number 0110 0000.

## SYSTem:BEEPer[:IMMediate] (No Query Form)

This command causes the waveform generator to emit an audible tone.

**Group** System

**Syntax** SYSTem:BEEPer[:IMMediate] [<frequency>[,<time>[,<volume>]]]



**Arguments** The following parameters are available, but are ignored:

<frequency> The pitch of audible tones  
 <time> The duration of audible tones  
 <volume> The volume of audible tones

**Examples** SYSTem:BEEPer:IMMediate  
 turns on a beep sound.

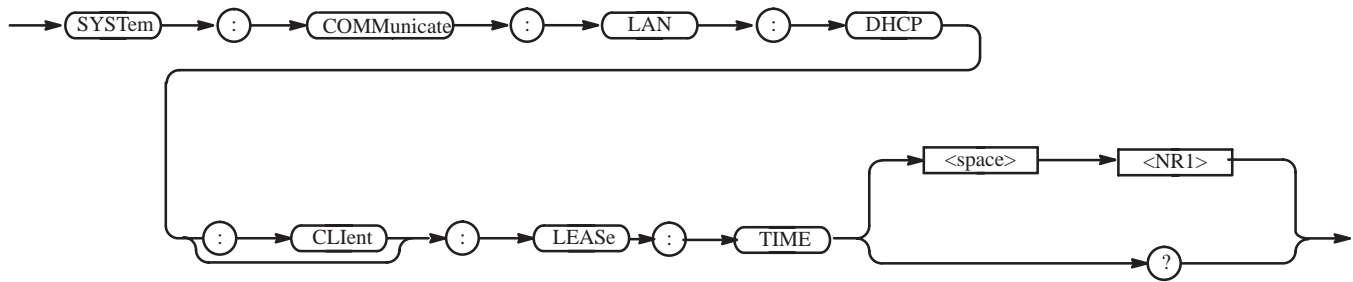
## SYSTem:COMMunicate:LAN:DHCP[:CLient]:LEASe:TIME (?)

*AWG400 series Only*

This command sets the IP address lease time of the DHCP client function.

**Group** System

**Syntax** SYSTem:COMMunicate:LAN:DHCP[:CLient]:LEASe:TIME <NR1>  
 SYSTem:COMMunicate:LAN:DHCP[:CLient]:LEASe:TIME?



**Arguments** <NR1> – lease time. The range is 30 to 86400, unit is "s".  
 At \*RST, this value is set to 28800.

**Examples** SYSTEM:COMMunicate:LAN:DHCP:CLient:LEASe:TIME 7200  
 sets the lease time to 7200sec.

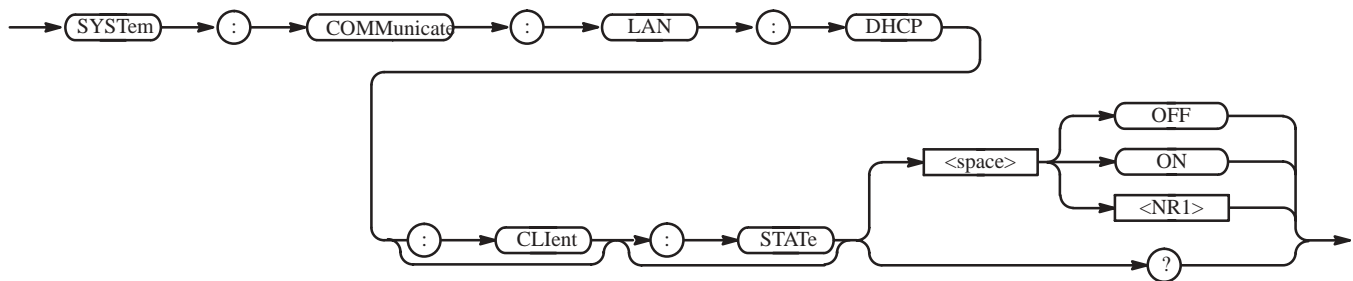
## SYSTEM:COMMunicate:LAN:DHCP[:CLient][:STATE] (?)

*AWG400 series Only*

This command turns on or off the DHCP client function.

**Group** System

**Syntax** SYSTEM:COMMunicate:LAN:DHCP[:CLient][:STATE] { ON | OFF |<NR1>}  
 SYSTEM:COMMunicate:LAN:DHCP[:CLient][:STATE]?



**Arguments** OFF or <NR1> = 0 turns off the DHCP client function.  
 ON or <NR1> ≠ 0 turns on the DHCP client function.  
 \*RST has no effect on the value.

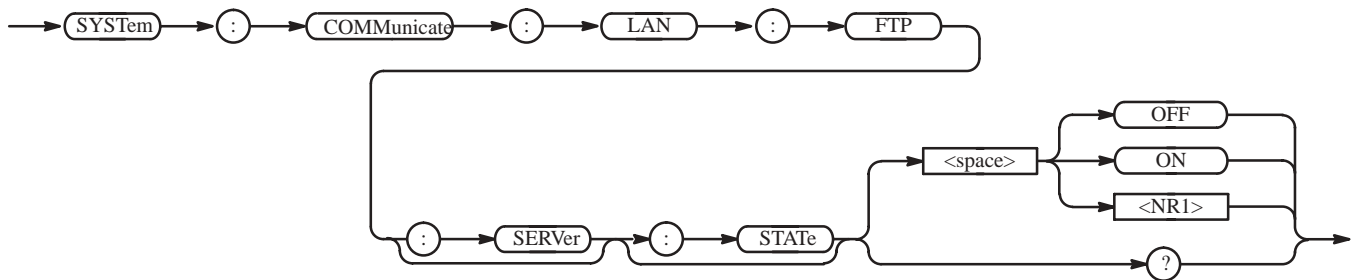
**Examples**    `SYSTem:COMMUnicate:LAN:DHCP:CLient:STATe ON`  
sets the DHCP client function to on.

## **SYSTem:COMMUnicate:LAN:FTP[:SERVer][:STATe] (?)**

This command turns on or off the FTP (File Transfer Protocol) server function.

**Group**    System

**Syntax**    `SYSTem:COMMUnicate:LAN:FTP[:SERVer][:STATe] { ON | OFF |<NR1>}`  
`SYSTem:COMMUnicate:LAN:FTP[:SERVer][:STATe]?`



**Arguments**    OFF or <NR1> = 0 turns off the FTP server function.  
ON or <NR1> ≠ 0 turns on the FTP server function.  
\*RST has no effect on the value.

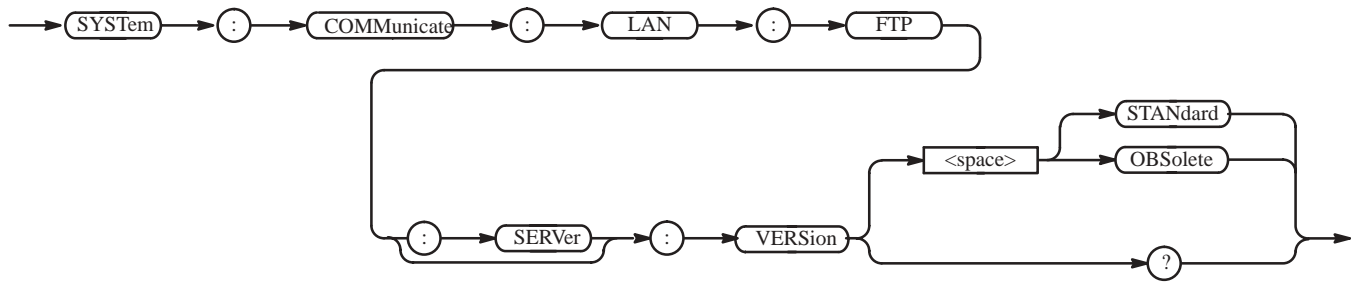
**Examples**    `SYSTem:COMMUnicate:LAN:FTP:SERVer:STATe ON`  
sets the FTP server function on.

## **SYSTem:COMMUnicate:LAN:FTP[:SERVer]:VERSIon (?)**

This command changes the version of the FTP (File Transfer Protocol) server.

**Group**    System

**Syntax**    `SYSTem:COMMUnicate:LAN:FTP[:SERVer]:VERSIon[:STATe] { STANDARD | OBSolete}`  
`SYSTem:COMMUnicate:LAN:FTP[:SERVer]:VERSIon?`



**Arguments** STANdar change the FTP server version to standard.  
 OBSolete change the FTP server version to obsolete(program version 2.x).  
 \*RST has no effect on the value.

**Examples** SYSTEM:COMMunicate:LAN:FTP:SERVer:VERsion OBSolete  
 sets the FTP server version to obsolete.

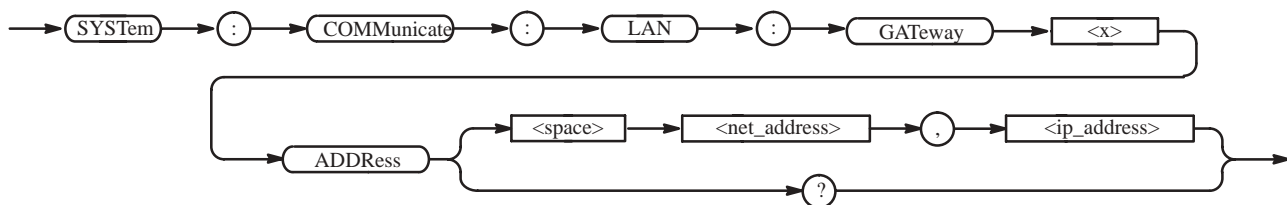
## SYSTEM:COMMunicate:LAN:GATeway[1|2|3]:ADDRess (?)

This command sets the IP address of the gateway when you communicate with the AWG400/500/600 Series Arbitrary Waveform Generator from anywhere other than the local network segment.

**Group** System

**Syntax** SYSTEM:COMMunicate:LAN:GATeway[1|2|3]:ADDRess  
 <net\_address>,<ip\_address>

SYSTEM:COMMunicate:LAN:GATeway[1|2|3]:ADDRess?



**Arguments** <net\_address>::=<string> is the network address.  
 <ip\_address>::=<string> is the IP address of the gateway.  
 \*RST has no effect on the value.

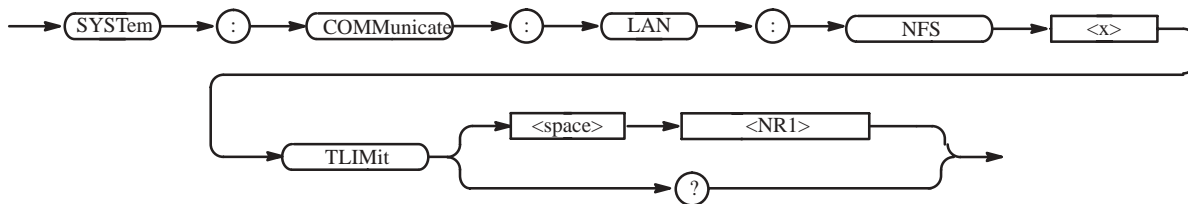
**Examples**    `SYSTem:COMMunicate:LAN:GATeway1:ADDRess "91.0.0.0","90.0.0.2"`  
sets the IP address of gateway 1 to 90.0.0.2 on the net 91.0.0.0.

## SYSTem:COMMunicate:LAN:NFS:TLIMit (?)

This command sets the NFS timeout.

**Group**    System

**Syntax**    `SYSTem:COMMunicate:LAN:NFS:TLIMit <NR1>`  
`SYSTem:COMMunicate:LAN:NFS:TLIMit?`



**Arguments**    <NR1> is the NFS timeout. The range is 25 to 300, the unit is “sec”.  
At \*RST, the parameter is set to 300.

**Examples**    `SYSTem:COMMunicate:LAN:NFS:TLIMit 60`  
sets the NFS timeout to 60 sec.

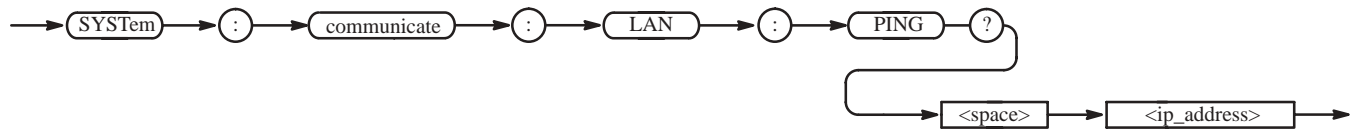
## SYSTem:COMMunicate:LAN:PING? (Query Only)

This command executes the ping test, and sends the ICMP ECHO\_REQUEST packet to a specified IP address.

**Group**    System

**Related Commands**    `SYSTem:COMMunicate:LAN:GATeway:ADDRess`  
`SYSTem:COMMunicate:LAN[:SELF]:ADDRess`

**Syntax**    `SYSTem:COMMunicate:LAN:GATeway:PING? <ip_address>`



**Arguments** <ip\_address>::=<string> is the IP address to be tested.

**Returns** <NR1>=1 indicates there was a response to the ECHO\_REQUEST packet.  
 <NR1>=0 indicates there was no response to the ECHO\_REQUEST packet.

**Examples** SYSTEM:COMMUNICATE:LAN:PING? "2.199.55.1"  
 might return a 1, indicating that there was a response from the host 2.199.55.1.

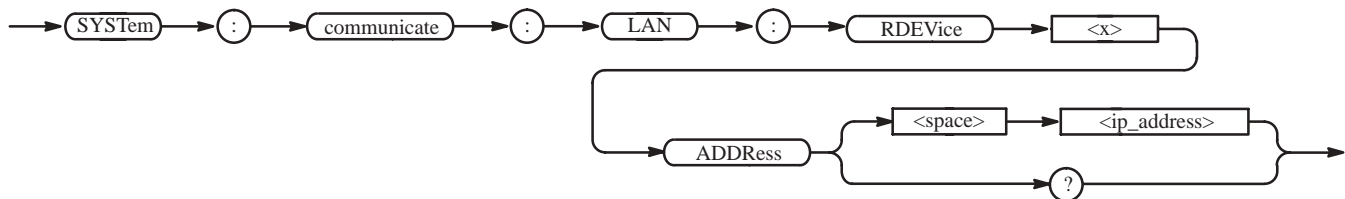
## SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:ADDRESS (?)

This command sets the IP address of the remote host. The host corresponds to “NET<x>” in the menu display. (You can change this name by using the SYSTEM:COMMUNICATE:LAN:RDEVICE<x>:NAME command.)

**Group** System

**Related Commands** SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:FSYSTEM  
 SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:NAME

**Syntax** SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:ADDRESS <ip\_address>  
 SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:ADDRESS?



**Arguments** <ip\_address>::=<string> is the IP address of the remote host.  
 \*RST has no effect on the value.

**Examples** SYSTEM:COMMUNICATE:LAN:RDEVICE1:ADDRESS "2.199.55.1"  
 sets the IP address of the remote host 1 (NET1) to 2.199.55.1.



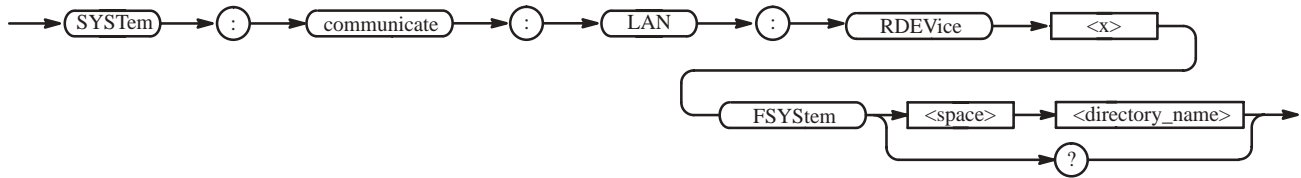
## SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:FSYStem (?)

This command sets the mount directory on a specified remote host.

**Group** System

**Related Commands** SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:ADDRes

**Syntax** SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:FSYStem <directory\_name>  
SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:FSYStem?



**Arguments** <directory\_name>::=<string> is the mount directory on the remote host.

\*RST has no effect on the value.

**Examples** SYSTem:COMMunicate:LAN:RDEvice1:FSYStem "/AWG/SAMPLE"  
sets the mount directory to /AWG/SAMPLE on the remote host 1 (NET1).

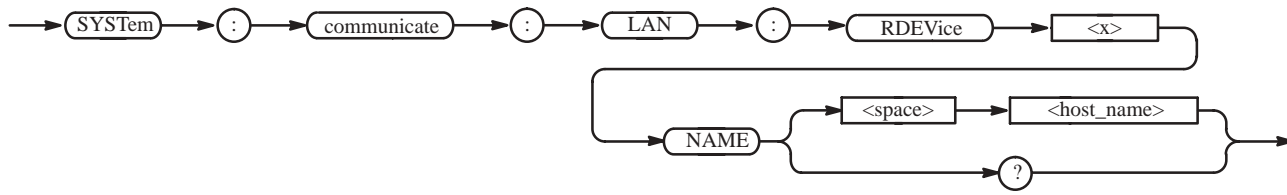
## SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:NAME (?)

This command sets the name of a specified remote host. The factory default name is "NET<x>", which may be displayed on the waveform generator menu. You can change the displayed host name using this command.

**Group** System

**Related Commands** SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:ADDRes

**Syntax** SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:NAME <host\_name>  
SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:NAME?



**Arguments** <host\_name>::=<string> is the name of the remote host. The name must be ten characters or less.

\*RST has no effect on the parameter.

**Examples** SYSTEM:COMMUNICATE:LAN:RDEVICE1:NAME "HOST1"  
sets the name of the remote host 1 to HOST1.

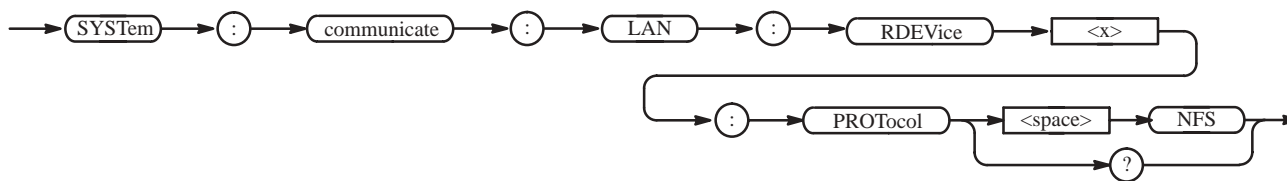
### SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:PROTOCOL (?)

This command selects the protocol of communication with the remote host. For this application, however, the protocol is fixed to NFS (Network File System), and this command exists only for compatibility.

**Group** System

**Related Commands** SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:ADDRESS

**Syntax** SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:PROTOCOL NFS  
SYSTEM:COMMUNICATE:LAN:RDEVICE[1|2|3]:PROTOCOL?



**Arguments** NFS selects the NFS protocol. This is fixed.

\*RST has no effect on this parameter.

**Examples** SYSTEM:COMMUNICATE:LAN:RDEVICE1:PROTOCOL NFS  
selects the NFS protocol.

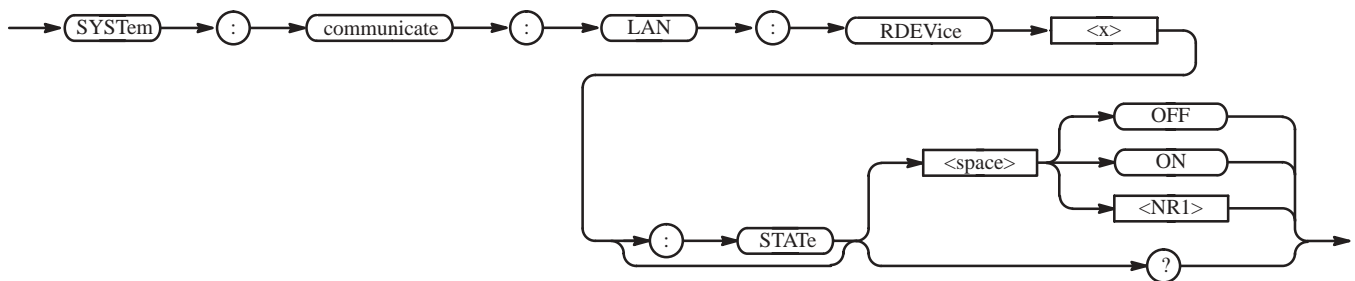
## SYSTem:COMMunicate:LAN:RDEvice[1|2|3][:STATe] (?)

This command turns the LAN communication (Remote host's directory mount of NFS protocol) on or off, using the remote host.

**Group** System

**Related Commands** SYSTem:COMMunicate:LAN:RDEvice[1|2|3]:ADDRESS

**Syntax** SYSTem:COMMunicate:LAN:RDEvice[1|2|3][:STATe] { ON | OFF | <NR1> }  
SYSTem:COMMunicate:LAN:RDEvice[1|2|3][:STATe]?



**Arguments** OFF or <NR1> = 0 turns off the LAN communication with the remote host.  
ON or <NR1> ≠ 0 turns on the LAN communication with the remote host.  
\*RST has no effect on the value.

**Examples** SYSTem:COMMunicate:LAN:RDEvice1:STATe ON  
turns on LAN communication with the remote host.

## SYSTem:COMMunicate:LAN[:SELF]:ADDRESS (?)

This command sets the IP address of the AWG400/500/600 Series Arbitrary Waveform Generator.

---

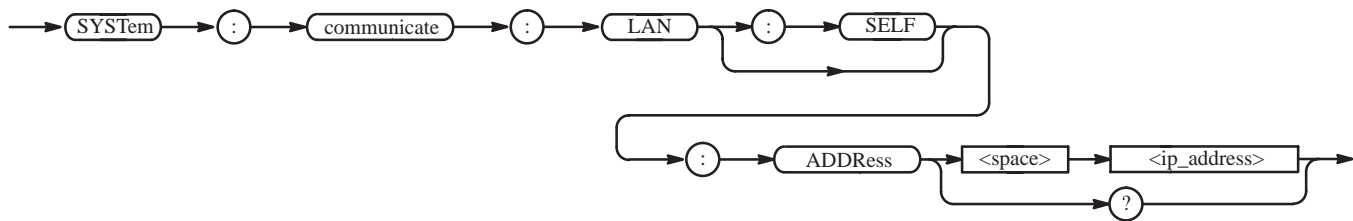
**NOTE.** You must set the IP address of the AWG400/500/600 Series Arbitrary Waveform Generator in order to use its LAN functions. If you specify "" (null) for the IP address, the LAN functions do not work.

---

**Group** System

**Related Commands** SYSTem:COMMunicate:LAN[:SELF]:SMASK

**Syntax** SYSTem:COMMunicate:LAN[:SELF]:ADDRess <ip\_address>  
 SYSTem:COMMunicate:LAN[:SELF]:ADDRess?



**Arguments** <ip\_address>::=<string> is the IP address of the AWG400/500/600 Series Arbitrary Waveform Generator.

\*RST has no effect on the value.

**Examples** SYSTem:COMMunicate:LAN:SELF:ADDRess "2.199.55.1"  
 sets the IP address of the AWG400/500/600 Series Arbitrary Waveform Generator.

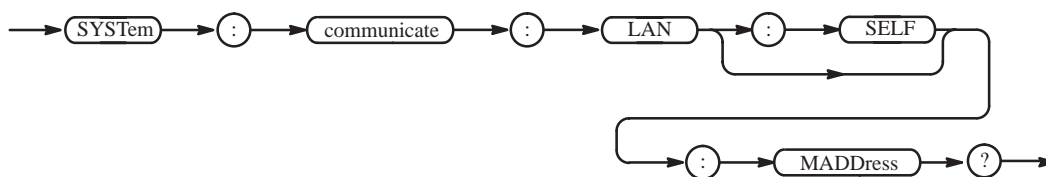
## SYSTem:COMMunicate:LAN[:SELF]:MADDRess? (Query Only)

### AWG400 series Only

This command returns the MAC address.

**Group** System

**Syntax** SYSTem:COMMunicate:LAN[:SELF]:MADDRess?



**Arguments** <string> is the MAC address.

**Examples**    `SYSTem:COMMunicate:LAN:SELF:MADdress?`  
might return the following response:

"XX:XX:XX:XX:XX:XX"

This response indicates the MAC address.

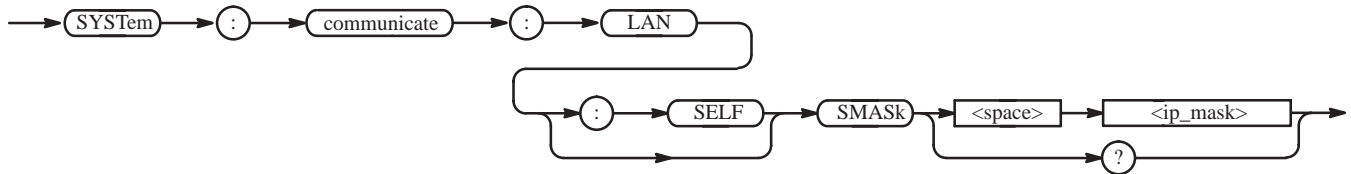
## **SYSTem:COMMunicate:LAN[:SELF]:SMASK (?)**

This command sets the subnet mask of the AWG400/500/600 Series Arbitrary Waveform Generator.

**Group**        System

**Related Commands**    `SYSTem:COMMunicate:LAN[:SELF]:ADdRes`

**Syntax**        `SYSTem:COMMunicate:LAN[:SELF]:SMASK <ip_mask>`  
`SYSTem:COMMunicate:LAN[:SELF]:SMASK?`



**Arguments**    `<ip_mask>::=<string>` is the subnet mask of the AWG400/500/600 Series Arbitrary Waveform Generator.

\*RST has no effect on the value.

**Examples**        `SYSTem:COMMunicate:LAN:SELF:SMASK "255.0.0.0"`  
sets the subnet mask to 255.0.0.0 for the AWG400/500/600 Series Arbitrary Waveform Generator.

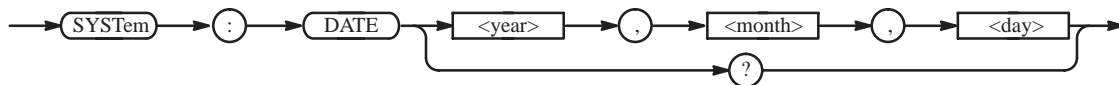
## SYSTem:DATE (?)

This command sets the date for the AWG400/500/600 Series Arbitrary Waveform Generator operating system.

**Group** System

**Syntax** SYSTem:DATE <year>,<month>,<day>

SYSTem:DATE?



**Arguments** <year>::=<NRf> must be entered as a four-digit number.

<month>::=<NRf> ranges 1 to 12.

<day>::=<NRf> ranges 1 to 31.

The range (AWG400 series) :2001.1.1 – 2099.12.31

The range (AWG500/600 series):1997.1.1 – 2095.12.31

<NRf> is rounded to the nearest integer.

\*RST has no effect on the value.

**Examples** SYSTem:DATE 2001,10,31  
sets the date.

## SYSTem:ERRor? (Query Only)

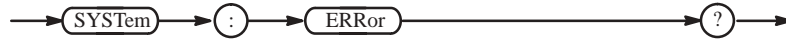
### AWG500/600 series Only

This command retrieves and returns error data from the Error and Event Queue. For more details, refer to the *Status and Event* section of this manual.

**Group** System

**Related Commands** STATus:QUEue[:NEXT]?

**Syntax** SYSTem:ERRor?



**Arguments** None

**Returns** <error/event\_number>,  
 "<error/event\_description>[;<device\_dependent\_info>]"  
 where:  
 <error/event\_number> is an integer between -32768 and 32767.  
 0 indicates that no error or event has occurred.  
 Positive values are error/event numbers determined by this instrument.  
 Negative values are error/event numbers reserved in SCPI standards.  
 <error/event\_description> is a message relating to the error/event number.  
 <device\_dependent\_info> is more detailed information relating to the error/event number.

**Examples** SYSTem:ERRor:NEXT?  
 might return the following response:  
 -102,"Syntax error;possible invalid suffix - :SOUR:FREQ 2V"  
 This response indicates that the unit is invalid.

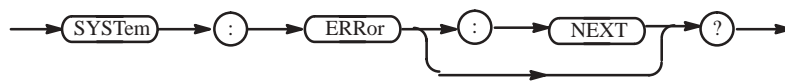
## SYSTem:ERRor[:NEXT]? (Query Only)

### AWG400 series Only

This command retrieves and returns error data from the Error and Event Queue. For more details, refer to the *Status and Event* section of this manual.

**Group** System

**Syntax** SYSTem:ERRor[:NEXT]?



**Arguments** None

**Returns** <error/event\_number>,  
 "<error/event\_description>[;<device\_dependent\_info>]"  
 where:  
 <error/event\_number> is an integer between -32768 and 32767.  
 0 indicates that no error or event has occurred.  
 Positive values are error/event numbers determined by this instrument.  
 Negative values are error/event numbers reserved in SCPI standards.  
 <error/event\_description> is a message relating to the error/event number.  
 <device\_dependent\_info> is more detailed information relating to the error/event number.

**Examples** SYSTem:ERRor:NEXT?  
 might return the following response:

-102,"Syntax error;possible invalid suffix - :SOUR:FREQ 2V"

This response indicates that the unit is invalid.



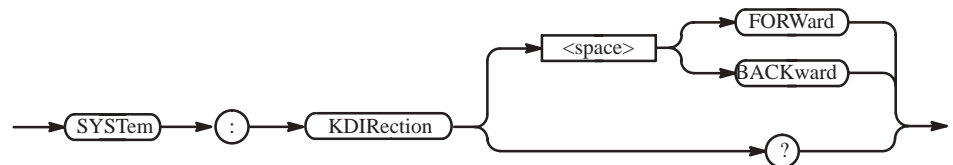
## SYSTem:KDIRrection (?)

This command determines the direction the cursor moves in response to the general purpose knob.

**Group** System

**Syntax** SYSTem:KDIRrection { FORWard | BACKward }

SYSTem:KDIRrection?



**Arguments** FORWard means the cursor moves to the right when the general purpose knob turns clockwise.

BACKward means the cursor moves to the left when the general purpose knob turns clockwise.

At \*RST, the parameter is set to FORWard.

**Examples** SYSTem:KDIRrection BACKward  
makes the cursor move to the left when you turn the general purpose knob clockwise.

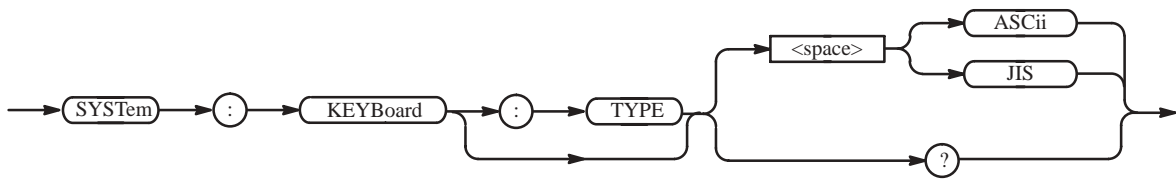
## SYSTem:KEYBoard[:TYPE] (?)

This command selects the type of keyboard that connects to the AWG400/500/600 Series Arbitrary Waveform Generator.

**Group** System

**Syntax** SYSTem:KEYBoard[:TYPE] { ASCii | JIS }

SYSTem:KEYBoard[:TYPE]?



**Arguments** ASCi i selects the ASCII 101-key keyboard.  
 JIS selects the JIS 106-key keyboard.  
 At \*RST, the parameter is set to ASCi i.

**Examples** SYSTem:KEYBoard:TYPE JIS  
 selects the JIS 106-key keyboard.

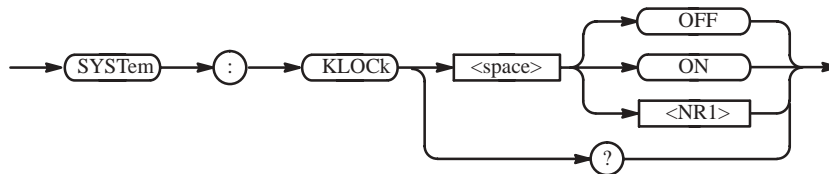
## SYSTem:KLOCK (?)

This command locks or unlocks the front panel and keyboard. Use this command to disable manual operation while the waveform generator is being controlled externally. If the front panel and keyboard are not explicitly locked out using this command, the waveform generator accepts input from both the external controller and the front panel and keyboard. Push CLEAR MENU(front-panel) twice to change from lock to unlock.

**Group** System

**Syntax** SYSTem:KLOCK { ON | OFF | <NR1> }

SYSTem:KLOCK?



**Arguments** OFF or <NR1> = 0 unlocks controls of the front panel and keyboard.  
 ON or <NR1> ≠ 0 locks controls of the front panel and keyboard.  
 \*RST has no effect on the parameter.

**Returns** <NR1> = 0 indicates the front panel and keyboard are unlocked.

<NR1> = 1 indicates the front panel and keyboard are locked.

**Examples**    SYSTem:KLOCK ON  
locks the front panel and keyboard.

SYSTem:KLOCK?  
might return 1, which indicates that the front panel and keyboard are locked.

## SYSTem:SECurity:IMMediate (No Query Form)

This command immediately destroys all waveform generator data and settings. Current settings are initialized to their \*RST values.

---

**NOTE.** This command erases all information on the internal hard disk (“MAIN”).

---

**Group**        System

**Related Commands**    \*RST

**Syntax**        SYSTem:SECurity:IMMediate



**Arguments**    None.

**Examples**        SYSTem:SECurity:IMMediate  
destroys all waveform generator data and settings.

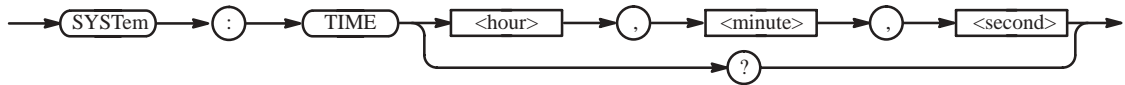
## SYSTem:TIME (?)

This command sets the internal clock.

**Group**        System

**Related Commands**    SYSTem:DATE

**Syntax**    SYSTem:TIME <hour>,<minute>,<second>  
 SYSTem:TIME?



**Arguments**    <hour>,<minute>,<second>  
 <hour>::=<NRf> ranges 0 to 23.  
 <minute>::=<NRf> ranges 0 to 59.  
 <second>::=<NRf> ranges 0 to 59.  
 It is always rounded to the nearest integer.

**Examples**    SYSTem:TIME 11,23,58  
 sets the time.

## SYSTem:UPTime? (Query Only)

This command queries how much time has elapsed from the generator power-on.

**Group**    System

**Syntax**    SYSTem:UPTime?



**Returns**    <hour>,<minute>,<second>  
 where  
 <hour>::=<NR1> ranges 0 to 23.  
 <minute>::=<NR1> ranges 0 to 59.  
 <second>::=<NR1> ranges 0 to 59.

**Examples**    SYSTem:UPTime?  
 might return 3,18,52, which indicates 3 hours 18 minutes and 52 seconds have elapsed after you powered on the waveform generator.

## SYSTem:VERSion? (Query Only)

This command returns the SCPI version number with the waveform generator complies.

**Group** System

**Syntax** SYSTem:VERSion?



**Returns** <NR2>:::=YYYY.V  
where YYYY represents the year version and V represents an approved revision number for that year.

**Examples** SYSTem:VERSion?  
might return 1999.0.

## \*TRG (No Query Form)

This command generates a trigger event. This command is equivalent to the TRIGger[:SEquence] [:IMMEDIATE] command or pressing the FORCE TRIGGER button on the front panel.

**Group** Trigger

**Related Commands** TRIGger[:SEquence] [:IMMEDIATE]

**Syntax** \*TRG



**Arguments** None

**Examples** \*TRG  
generates a trigger event.

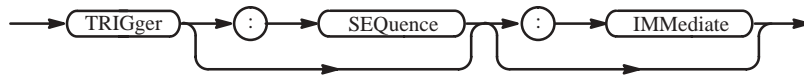
## TRIGger[:SEquence][:IMMediate] (No Query Form)

This command generates a trigger event. This command is equivalent to the \*TRG command or pressing the FORCE TRIGGER button on the front panel.

**Group** Trigger

**Related Commands** \*TRG

**Syntax** TRIGger[:SEquence] [:IMMediate]



**Arguments** None

**Examples** TRIGger:SEquence:IMMediate  
generates the trigger event.

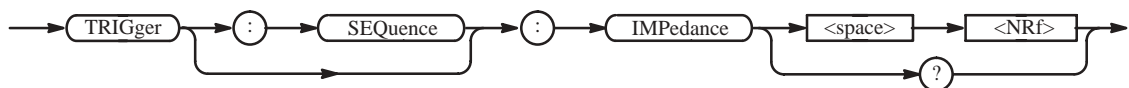
## TRIGger[:SEquence]:IMPedance (?)

This command selects the impedance of the external trigger input.

**Group** Trigger

**Syntax** TRIGger[:SEquence]:IMPedance <NRf>

TRIGger[:SEquence]:IMPedance?



**Arguments** <NRf> is 50 (50  $\Omega$ ) or 1e3 (1 k $\Omega$ ).

At \*RST, the value is set to 1 k $\Omega$ .

**Examples** TRIGger:SEquence:IMPedance 50  
selects 50  $\Omega$  impedance for the external trigger input.

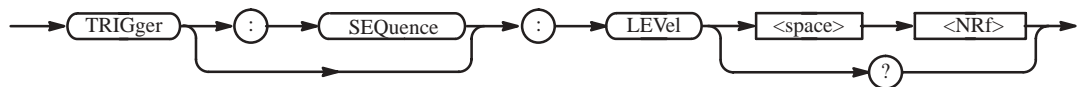
## TRIGger[:SEQuence]:LEVEl (?)

This command sets the trigger level on the selected SOURCE.

**Group** Trigger

**Related Commands** TRIGger[:SEQuence]:SOURce

**Syntax** TRIGger[:SEQuence]:LEVEl <NRf>  
TRIGger[:SEQuence]:LEVEl?



**Arguments** <NRf> is the trigger level. The range is  $-5.0$  V to  $+5.0$  V, in  $0.1$  V steps.  
At \*RST, the value is set to  $1.4$  V.

**Examples** TRIGger:SEQuence:LEVEl 200mV  
sets the trigger level to  $200$  mV.

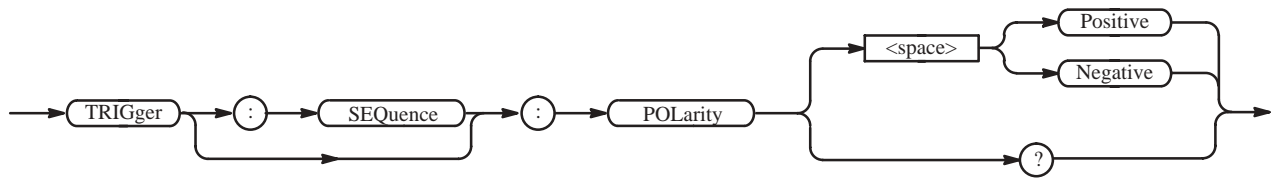
## TRIGger[:SEQuence]:POLarity (?)

This command selects the polarity relative to the trigger level that is required to activate the gate signal. This command is effective only when the waveform generator is in the gated mode.

**Group** Trigger

**Related Commands** AWGControl:RMODE, TRIGger[:SEQuence]:LEVEl

**Syntax** TRIGger[:SEQuence]:POLarity { Positive | Negative }  
TRIGger:POLarity?



**Arguments** Positive means the gate signal is activated when the external trigger signal is greater (more Positive) than the trigger level.

Negative means the gate signal is activated when the external trigger signal is less (more Negative) than the trigger level.

At \*RST, the parameter is set to Positive.

**Examples** TRIGger[:SEQuence]:POLarity Negative selects the Negative polarity.

## TRIGger[:SEQuence]:SLOPe (?)

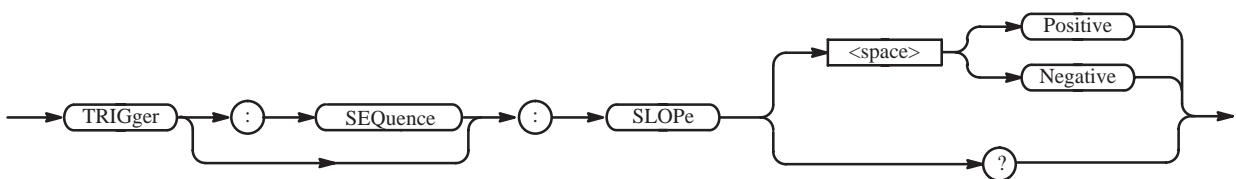
This command determines whether the event occurs on the the rising edge or falling edge of the external trigger signal.

**Group** Trigger

**Related Commands** TRIGger[:SEQuence]:SOURce

**Syntax** TRIGger[:SEQuence]:SLOPe { Positive | Negative }

TRIGger[:SEQuence]:SLOPe?



**Arguments** Positive means the event occurs on the rising edge of the external trigger signal.

Negative means the event occurs on the falling edge of the external trigger signal.

At \*RST, the parameter is set to Positive.



**Examples** TRIGger:SEquence:SLOPe Negative  
selects the Negative slope.

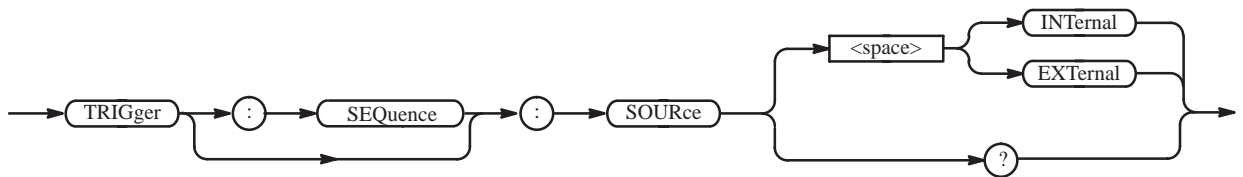
## TRIGger[:SEquence]:SOURce (?)

This command selects the trigger source.

**Group** Trigger

**Related Commands** TRIGger[:SEquence]:LEVel, TRIGger[:SEquence]:POLarity,  
TRIGger[:SEquence]:SLOPe, TRIGger[:SEquence]:TIMER

**Syntax** TRIGger[:SEquence]:SOURce { INTernal | EXTernal }  
TRIGger[:SEquence]:SOURce?



**Arguments** INTernal selects the internal clock as the trigger source.

EXTernal selects the external trigger input as the trigger source.

At \*RST, the parameter is set to EXTernal.

**Examples** TRIGger:SEquence:SOURce INTernal  
selects the internal clock as the trigger source.

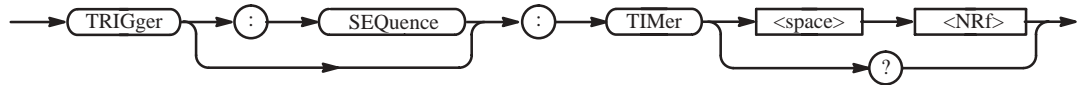
## TRIGger[:SEquence]:TIMER (?)

This command sets the period of the internal clock when you select the internal clock as the trigger source with the TRIGger[:SEquence]:SOURce command.

**Group** Trigger

**Related Commands** TRIGger[:SEquence]:SOURce

**Syntax** TRIGger[:SEquence]:TIMer <NRf>  
 TRIGger[:SEquence]:TIMer?



**Arguments** <NRf> is the internal trigger rate. The range is 1.0  $\mu$ s to 10.0 s.  
 At \*RST, this value is set to 100 ms.

**Examples** TRIGger:SEQuence:TIMer 5ms  
 sets the internal trigger rate to 5 ms.

### \*TST? (Query Only)

This command performs the selftest and returns the results. If an error is detected during selftest, execution is stopped immediately.

---

**NOTE.** This command takes several minutes to complete the self test, the waveform generator will not respond to any commands and queries during this time.

---

**Group** Diagnostic

**Related Commands** \*CAL?, CALibration[:ALL], DIAGnostic[:IMMediate]

**Syntax** \*TST?



**Arguments** None

**Returns** <NR1>

0 Terminated without error.  
 -330 Selftest failed.

**Examples** \*TST?  
might return -330 indicating the selftest failed.

## \*WAI (No Query Form)

This command prevents the waveform generator from executing further commands or queries until all pending operations finish.

In AWG400/500/600 series and in this application, all commands are designed to be executed in the order in which they are sent from the external controller. The \*WAI command is included to ensure compliance with the SCPI standard. You do not need to use this command.

**Group** Synchronization

**Related Commands** \*OPC

**Syntax** \*WAI

→  →

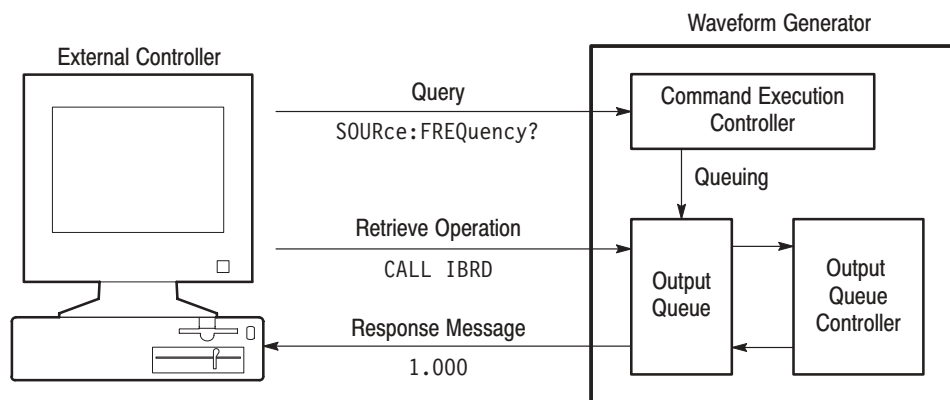
**Arguments** None

**Examples** \*WAI  
prevents the execution of any commands or queries until all pending operations complete.



## Retrieving Response Messages

When a query command is sent from the external controller, the waveform generator places a response message on the output queue. To retrieve this response message you must perform a retrieval operation through the external controller. For example, you can call the IBRD subroutine with the National Instruments drivers for the GPIB interface (see Figure 2–9).



**Figure 2–9: Retrieving response messages**

Before a response message is placed in the output queue, the previous response message, if any, is deleted. Thus, if a second query occurs before the first response message is retrieved, the first response message will be lost.

The SBR (status byte register) MAV bit can be used to check the response message queuing state. Refer to the *Status and Events* section in this manual for more information about the output queue, SBR, and control methods.



# Data Transfer

You can transfer data between the waveform generator and external devices through the GPIB and Ethernet LAN interface. This section describes required data formats and transfer procedures.

## Data File

The waveform generator uses these file types:

- The Waveform file contains waveform data in single precision floating point format.
- The Pattern file contains waveform data in binary format.
- The Sequence file defines the output sequence.
- The Equation file uses numeric formulas to describe the output waveform.
- The Code Convert file contains the Code Convert Table.

During front panel operation, the waveform generator creates these files automatically; when you remotely operate the waveform generator, you must create these files through editing or programming according to the formats described in the topics that follow.

## About Waveform and Pattern Files

To output a wave form you can load both the Waveform and Pattern file. When you load a Waveform file, it is converted and stored to waveform memory with these digital patterns:

- For the AWG400 series, a 16 bit digital pattern.
- For the AWG500 series, a 10 bit digital pattern.
- For the AWG600 series, an 8 bit digital pattern.

The instrument stores data in the Pattern file to waveform memory without conversion.

The difference between these two files is the internal format and the editor. The Waveform file format is composed of a 4-byte Little Endian format specified in IEEE488.2 floating point numbers, and 1-byte of marker data (see page 2–131 for format details). The Pattern file format is composed of 3-bytes (AWG400 series), or 2-bytes (AWG500/600 series), including data and markers (see page 2–132 for format details).

### Guidelines for using files

Following are some guidelines for choosing either the Waveform file or Pattern file to output waveforms.

- Select a Pattern file to shorten the transfer time when you do not need to perform further edits or operations in the instrument. Although both files are the same data length, the volume of the Pattern file is always less than the volume of the Waveform file.
- Use a Waveform file when you use waveform data to generate another waveform with a mathematical operation. The Waveform file format retains the data precision required for mathematical operations.

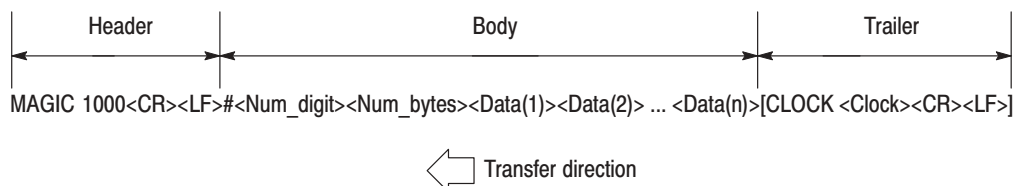
For more details about file formats, refer to *Data Transfer* section in one of these manuals:

*AWG410 & AWG420 & AWG430 Arbitrary Waveform Generator User Manual*  
*AWG510 & AWG520 Arbitrary Waveform Generator User Manual*  
*AWG610 Arbitrary Waveform Generator User Manual.*



**Waveform File** The Waveform file contains waveform data in single precision floating-point numbers and marker data.

**File Format.** The Waveform file consists of three main parts (see Figure 2–10).



**Figure 2–10: The Waveform file format**

<Waveform File> ::= <Header><Body> [<Trailer>]

where:

<Header> ::= MAGIC<space>1000<CR><LF>

<Body> ::= #<Num\_digits><Num\_bytes><Data(1)><Data(2)>...<Data(n)>

<Num\_digits> is the number of digits in <Num\_bytes>.

<Num\_bytes> is the byte count of the data that follows.

<Data(n)> ::= <Waveform><Marker>

<Waveform> is the single precision floating-point number of 4-byte Little Endian format specified in IEEE488.2. The full scale of the D/A converter of the waveform generator corresponds to –1.0 to 1.0.

<Marker> is one byte of marker data. The bit 0 (LSB) and bit 1 represent markers 1 and 2, respectively.

<Trailer> ::= CLOCK<space><Clock><CR><LF>

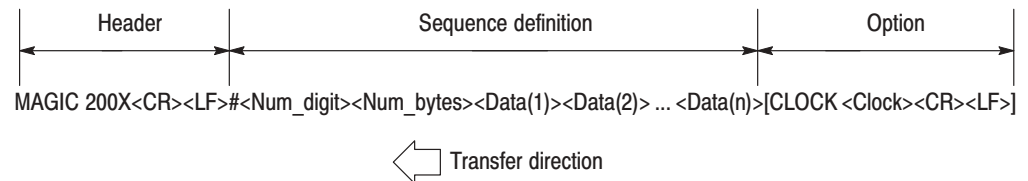
<Clock> is the value of the sample clock in ASCII.

**Example.** This example shows the contents of a Waveform file that contains two point data.

<pre> 4D 41 47 49 43 20 31 30 30 30 0D 0A 23 32 31 30 00 00 00 00 03 00 00 00 00 00 43 4C 4F 43 4B 20 31 2E 30 30 30 30 30 30 30 30 30 65 2B 30 38 0D 0A           </pre>	<pre> MAGIC 1000..#210 .....CLOCK 1.0000000000E+08 ..           </pre>
---	--

**Pattern File** The Pattern file contains waveform data in binary format.

**File Format.** The data consists of three main parts (see Figure 2–11).



**Figure 2-11: The Pattern File format**

<Pattern File>::=<Header><Body>[<Trailer>]

where:

AWG400 series:                   <Header>::=MAGIC<space>2003<CR><LF>

AWG500/600 series:           <Header>::=MAGIC<space>2000<CR><LF>

<Body>::=#<Num\_digits><Num\_bytes><Data(1)><Data(2)>...<Data(n)>

<Num\_digits> is the number of digits in <Num\_bytes>.

<Num\_bytes> is the byte count of the data that follows.

<Data(n)>

In the AWG400 series, this represents each data point in two bytes (16 bits: Data0 through Data15) and one byte (8 bits: Marker1 and Marker2), total three bytes (24 bits). Two bytes of Data0 through Data15, the low byte is transferred first.

Bits 0 through 15 are:

DIGITAL OUTPUT: used for D0 – D15 of ”CMOS Output Pod P4116”

ANALOG OUTPUT: used for Bit 0 (LSB) – Bit 15 (MSB)

Bits 1 and 2 of the next one byte are MARKER1 and MARKER2 of the rear panel.

In the AWG500/600 series, this represents each data point in two bytes (16 bits). The low byte is transferred first.

Bits 0 (LSB) – 9 are D0 – D9 of the rear panel. (In AWG600 series, Bits 2 (LSB) – 9 are D0 – D7).

Bits 13 and 14 are used for Markers 1 and 2, respectively.

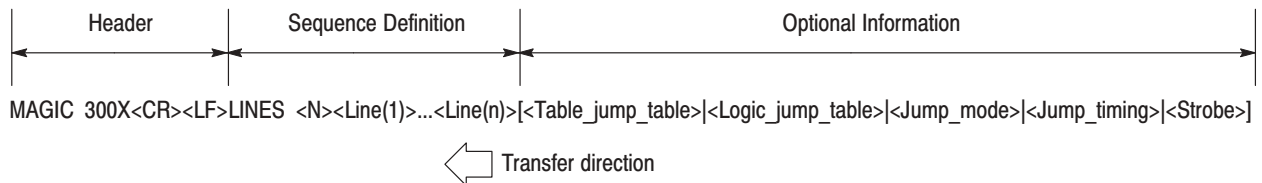
Bits 10 through 12, and 15 are unused and must be 0 (zero).

<Trailer> ::= CLOCK<space><Clock><CR><LF>

<Clock> is the value of the sample clock in ASCII.

**Sequence File** The Sequence file defines the output sequence in ASCII format.

**File Format.** The data consists of three main parts (see Figure 2–12).



**Figure 2–12: The Sequence File format**

```
<Sequence File>
::=<Header><Sequence Definition>[<Optional Information>]
```

where:

```
<Header>::=MAGIC<space>300x<CR><LF>
```

x=1, 2, or 3 represents the number of channels for which sequences are defined in the file.

```
<Sequence Definition>
::=LINES<space><N><Line(1)><Line(2)>...<Line(n)>
```

<N> is the number of lines that follow.

For MAGIC 3003:

```
<Line(n)>::=<CH1_file_name>,<CH2_file_name>,<CH3_file_name>,
<Repeat_count>
[,<Wait_trigger>[<Goto-1>[,<Logic_jump_target>]]]<CR><LF>
```

For MAGIC 3002:

```
<Line(n)>::=<CH1_file_name>,<CH2_file_name>,<Repeat_count>
[,<Wait_trigger>[<Goto-1>[,<Logic_jump_target>]]]<CR><LF>
```

For MAGIC 3001:

```
<Line(n)>::=<CH1_file_name>,<Repeat_count>
[,<Wait_trigger>[<Goto-1>[,<Logic_jump_target>]]]<CR><LF>
```

<CHx\_file\_name>::=<string> is the waveform or pattern file name for the specified channel.

<Repeat\_count>::=<NR1> is the repeat count for the line. 0 (zero) is infinity.

<Wait\_trigger>::=<NR1> specifies whether or not to wait for a trigger.  
<NR1>=0 is Off, ≠0 is On.

<Goto-1>::=<NR1> specifies whether or not to go to the next line.  
<NR1>=0 is Off, ≠0 is On.

<Logic\_jump\_target>::=<NR1> is line number for the Logic-Jump.  
0 is Off, -1 is Next, and -2 is Table-Jump. The default is Off.

#### <Optional Information>

```
::={ <Table_jump_table> | <Logic_jump_table> | <Jump_mode> |
<Jump_timing> | <Strobe> }
```

```
<Table_jump_table>
::=TABLE_JUMP<space><Jump_target(1)>,<Jump_target(2)>,
...<Jump_target(16)><CR><LF>
```

<Jump\_target(n)>::=<NR1> is the line number to the Table-Jump  
or 0 (Off). The default is Off.

```
<Logic_jump_table>
::=LOGIC_JUMP<space><Jump_on/off(1)>,<Jump_on/off(2)>,
<Jump_on/off(3)>,<Jump_on/off(4)><CR><LF>
```

<Jump\_on/off(n)>::=<NR1> sets the Logic-Jump on or off.  
<NR1>=0 is Off, 0 is On, and <0 is Ignore. The default is Ignore.

```
<Jump_mode>::=JUMP_MODE<space>{ LOGIC | TABLE | SOFTWARE }
<CR><LF>
```

sets the jump mode. The default is TABLE.

```
<Jump_timing>::=JUMP_TIMING<space>{ SYNC | ASYNC }<CR><LF>
```

sets the jump mode. The default is ASYNC.

```
<Strobe>::=STROBE<space><NR1><CR><LF>
```

determines whether or not to use the STROBE signal from the EVENT IN connector on the rear panel.  
<NR1>=0 is Off, ≠0 is On. The default is Off.

**Example.** This Sequence file contains two lines of sequence definitions for CH 1.

```
MAGIC 3001
LINES 2
"SAMPLE1.wfm",1,0,0,0
"SAMPLE3.wfm",1,0,0,0
TABLE_JUMP 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
LOGIC_JUMP -1,-1,-1,-1
JUMP_MODE TABLE
JUMP_TIMING ASYNC
STROBE 0
```

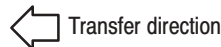
**Example.** This Sequence file contains two lines of sequence definitions for CH 1 and CH 2.

```
MAGIC 3002
LINES 2
"SAMPLE1.wfm", "SAMPLE2.wfm", 1, 0, 0, 0
"SAMPLE3.wfm", "SAMPLE4.wfm", 1, 0, 0, 0
TABLE_JUMP 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
LOGIC_JUMP -1, -1, -1, -1
JUMP_MODE TABLE
JUMP_TIMING ASYNC
STROBE 0
```

**Equation File** The Equation file describes the numerical formula that defines the output waveform in ASCII format.

**File Format.** The Equation file consists of ASCII characters (see Figure 2–13).

```
<Line(1)><CR><LF><Line(2)><CR><LF><Line(3)><CR><LF> ... <Line(n)><CR><LF>
```



**Figure 2-13: The Equation File format**

<Line(n)> represents each line of the Equation file. From single (') quotation marks to the end of the line is a comment. Characters enclosed in double (") quotation marks are a character string.

Detailed information about the functions and operators that can be used to describe the Equation file, refer to one of these manuals:

*AWG410 & AWG420 & AWG430 Arbitrary Waveform Generator User Manual*  
*AWG510 & AWG520 Arbitrary Waveform Generator User Manual*  
*AWG610 Arbitrary Waveform Generator User Manual.*

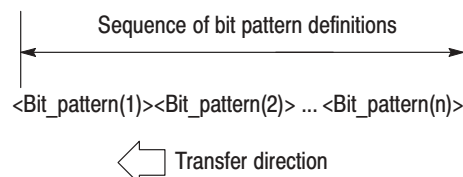
**Example.** This Equation file describes the log sweep waveform.

```
'frequency sweep sine (log)
clock=800e6
size=8800
k0=11e-6      'sweep period
k1=1e6        'starting frequency
k2=10e6       'ending frequency
k3=log(k2/k1)
"log_swp.wfm"=sin(2*pi*k1*k0/k3*(exp(k3*scale)-1))
```

**Code Convert File**

The Code Convert file is an ASCII text file that describes the Code Convert Table as displayed in the Edit menu.

**File Format.** The Code Convert file consists of bit pattern definitions (see Figure 2–14).



**Figure 2–14: The Code Convert File format**

```
<Code Convert File> ::= <Bit_pattern(1)><Bit_pattern(2)> ...
<Bit_pattern(n)>
```

where:

```
<Bit_pattern(n)> ::= [<Past Source>,<Current Source>,
<Next Source>,<Past Output>,<Output Code><CR><LF>]
```

<Past Source>, <Current Source>, <Next Source>, <Past Output>, and <Output Code> specifies the bit patterns in the Code Convert Table. The bit pattern is specified with “0”, “1”, and “–” (don’t care).

For more information about the Code Convert Table, refer to one of these manuals:

*AWG410 & AWG420 & AWG430 Arbitrary Waveform Generator User Manual*

*AWG510 & AWG520 Arbitrary Waveform Generator User Manual*

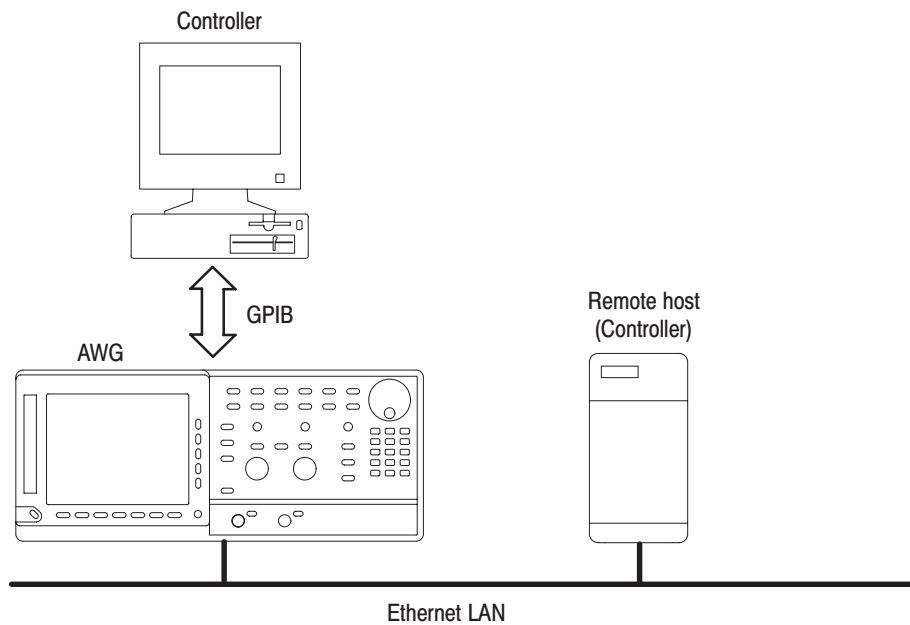
*AWG610 Arbitrary Waveform Generator User Manual.*

**Example.** This Code Convert file describes NRZI conversion.

```
-,0---,,0,0
-,0---,,1,1
-,0---,,0,1
-,0---,,1,0
```

## Data Transfer Procedures

Data can be loaded from the external controller to the waveform generator or from the waveform generator to the external controller through the GPIB interface, or through the Ethernet interface.



### External Device to Waveform Generator

Use the following command to transfer data from the external controller to the waveform generator:

```
MMEemory:DATA <file_name>,<data>
```

This command downloads <data> into the file <file\_name> on the internal hard disk, floppy disk, or the network drive. The default directory and mass memory device are specified by the MMEemory:CDIRECTory and MMEemory:MSIS commands respectively. The <data> is in IEEE488.2 block format.

For example, the following command string will load 2048 bytes of data to the file AWG1.

```
MMEemory:DATA "AWG1",#42048<data(1)><data(2)>...<data(2048)>
```



**Waveform Generator to  
External Device**

Use the following command to transfer data from the waveform generator to the external controller.

```
MMEemory:DATA? <file_name>
```

This command uploads the file <file\_name> on the internal hard disk, floppy disk, or the network drive. The response format is in IEEE488.2 block format.

For example, the following command string will upload the file FILE-AWG on the waveform generator to the external controller.

```
MMEemory:DATA? "FILE-AWG"
```





# Status and Events



# Status and Event Reporting

This section provides details about the status information and events the waveform generator reports.

## Status Reporting Structure

The waveform generator status reporting functions conform to IEEE-488.2 and SCPI standards. Use the status reporting function to check for instrument errors and to identify the types of events that have occurred on the instrument.

Figure 3–1 is a diagram of the instrument's status reporting function. The status reporting function is separated into three functional blocks:

- Standard/Event Status
- Operation Status
- Questionable Status

The operations processed in these three blocks are summarized in status bytes, which provide the error and event data.

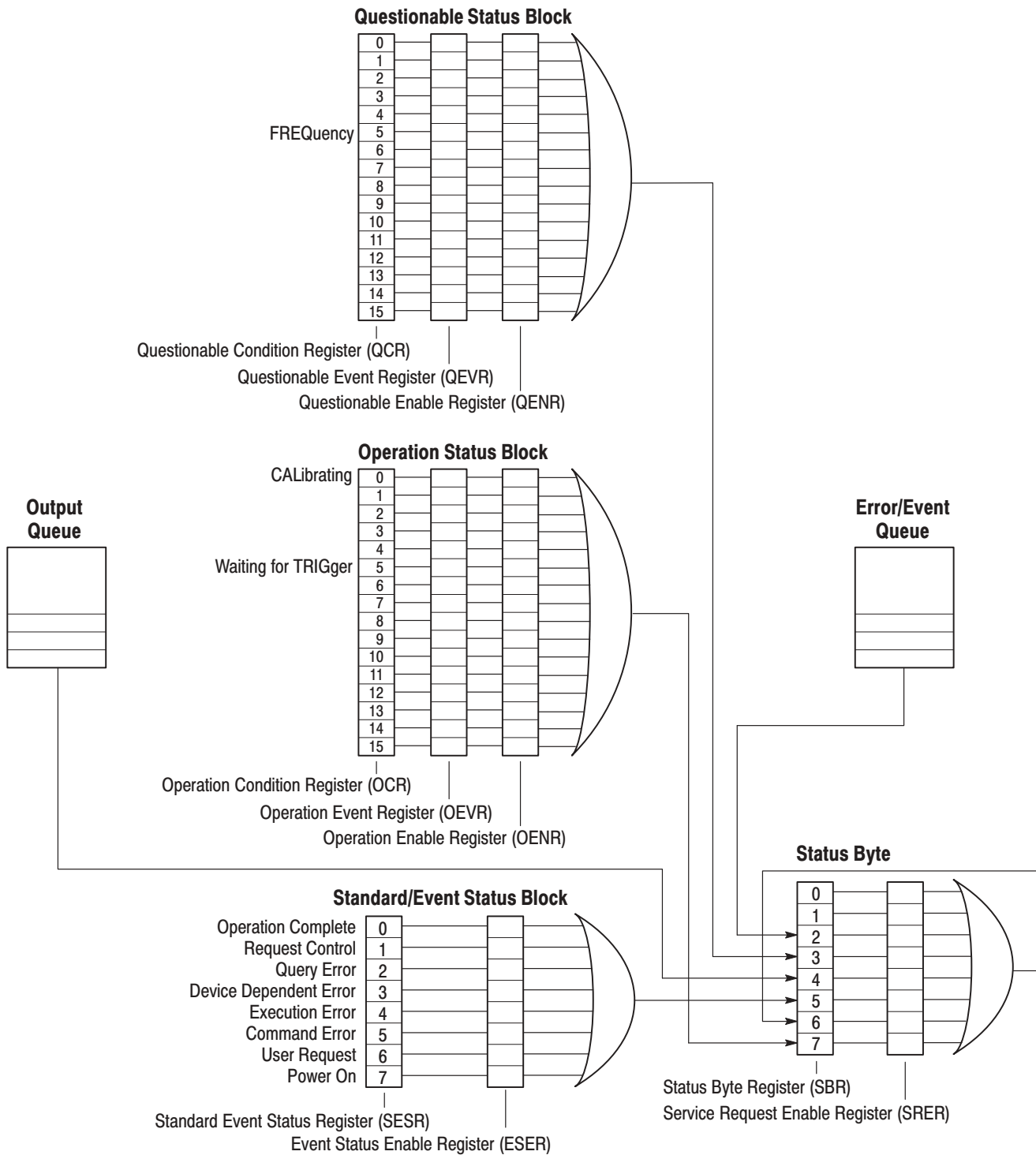


Figure 3-1: Error and Event handling process overview

**Standard/Event Status Block**

This block is used to report power on/off, command error, and command execution status.

The block has two registers: the Standard Event Status Register (SESR) and the Event Status Enable Register (ESER). Refer to the Standard/Event Status Block shown at the bottom of Figure 3-1 on page 3-2.

The SESR is an eight-bit status register. When an error or other type of event occurs on the instrument, the corresponding bit is set. You cannot write to this register. The ESER is an eight-bit enable register that masks the SESR. You can set this mask, and take AND with the SESR to determine whether or not the ESB bit in the Status Byte Register (SBR) should be set. Refer to *Event Status Enable Register (ESER)* on page 3-8, and *Standard Event Status Register (SESR)* on page 3-6, for the contents of these registers.

**Operation Status Block**

This block is used to report on the status of several operations being executed by the waveform generator.

The block is made up of three registers: the Operation Condition Register (OCR), the Operation Event Register (OEVR) and the Operation Enable Register (OENR). Refer to the Operation Status Block shown in the middle of Figure 3-1 on page 3-2.

When the instrument achieves a certain status, the corresponding bit is set to the OCR. You cannot write to this register. OCR bits that have changed from false (reset) to true (set) status are set in the OEVR. The function of the OENR is to mask the OEVR. You can set this mask and take AND with the OEVR to determine whether or not the OSS bit in the Status Byte Register (SBR) should be set. Refer to *Operation Condition Register (OCR)* on page 3-7, *Operation Event Register (OEVR)* on page 3-7, and *Operation Enable Register (OENR)* on page 3-9, for the contents of these registers.

**Questionable Status Block**

This block reports on the status of signals and data, such as the accuracy of entered data and signals generated by the instrument. The register configuration and process flow are the same as for the Questionable Status Block. Refer to *Questionable Condition Register (QCR)* on page 3-7, *Questionable Event Register (QEVR)* on page 3-8, and *Questionable Enable Register (QENR)* on page 3-9, for the contents of these registers.

## Registers

There are two main types of registers:

- **Status Registers:** store data relating to instrument status. These registers are set by the waveform generator.
- **Enable Registers:** determine whether to set events that occur in the instrument to the appropriate bits in the status registers and event queues. You can set this register.

## Status Registers

There are six types of status registers:

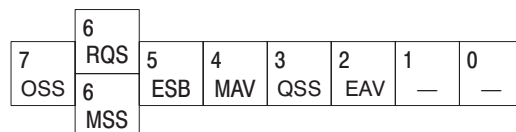
- Status Byte Register (SBR)
- Standard Event Status Register (SESR)
- Operation Condition Register (OCR)
- Operation Event Register (OEVR)
- Questionable Condition Register (QCR)
- Questionable Event Register (QEVVR)

Read the contents of these registers to determine errors and conditions.



**Status Byte Register (SBR)**

The SBR is made up of 8 bits. Bits 4, 5 and 6 are defined in accordance with IEEE Std 488.2-1987 (see Figure 3–2 and Table 3–1). These bits are used to monitor the output queue, SESR, and service requests, respectively. The contents of this register are returned when the \*STB? query is used.



**Figure 3–2: The Status Byte Register (SBR)**

**Table 3–1: SBR bit functions**

Bit	Function
7	Operation Summary Status (OSS).
6	RQS (Request Service)/MSS (Master Summary Status). When the instrument is accessed using the GPIB serial poll command, this bit is called the Request Service (RQS) bit and indicates to the controller that a service request has occurred (in other words, that the GPIB bus SRQ line is LOW). The RQS bit is cleared when serial poll ends.  When the instrument is accessed using the *STB? query, this bit is called the Master Summary Status (MSS) bit and indicates that the instrument has issued a service request for one or more reasons. The MSS bit is never cleared to 0 by the *STB? query.
5	Event Status Bit (ESB). This bit indicates whether or not a new event has occurred after the previous Standard Event Status Register (SESR) has been cleared or after an event readout has been performed.
4	Message Available Bit (MAV). This bit indicates that a message has been placed in the output queue and can be retrieved.
3	Questionable Summary Status (QSS).
2	Event Queue Available (EAV).
1–0	Not used

**Standard Event Status Register (SESR)**

The SESR is made up of 8 bits. Each bit records the occurrence of a different type of event, shown in Figure 3–3 and Table 3–2. The contents of this register are returned when the \*ESR? query is used.

7	6	5	4	3	2	1	0
PON	—	CME	EXE	DDE	QYE	—	OPC

**Figure 3-3: The Standard Event Status Register (SESR)**

**Table 3-2: SESR bit functions**

Bit	Function
7	Power On (PON). Indicates that the power to the instrument is on.
6	Not used.
5	Command Error (CME). Indicates that a command error has occurred while parsing by the command parser was in progress.
4	Execution Error (EXE). Indicates that an error occurred during the execution of a command. Execution errors occur for one of the following reasons: <ul style="list-style-type: none"> <li>■ A value designated in the argument is outside the allowable range of the instrument, or is in conflict with the capabilities of the instrument</li> <li>■ The command could not be executed properly because the conditions for execution differed from those essentially required</li> </ul>
3	Device-Specific Error (DDE). An instrument error has been detected.
2	Query Error (QYE). Indicates that a query error has been detected by the output queue controller. Query errors occur for one of the following reasons: <ul style="list-style-type: none"> <li>■ An attempt was made to retrieve messages from the output queue, despite the fact that the output queue is empty or in pending status.</li> <li>■ The output queue messages have been cleared despite the fact that they have not been retrieved.</li> </ul>
1	Not used.
0	Operation Complete (OPC). This bit is set with the results of the execution of the *OPC command. It indicates that all pending operations have been completed.

**Operation Condition Register (OCR)**

The OCR is made up of 16 bits, which record the occurrence of three types of events, shown in Figure 3–4 and Table 3–3.

15	14	13	12	11	10	9	8	7	6	5 TRIG	4	3	2	1	0 CAL
----	----	----	----	----	----	---	---	---	---	-----------	---	---	---	---	----------

**Figure 3–4: The Operation Condition Register (OCR)**

**Table 3–3: OCR bit functions**

Bit	Function
15 – 6	Not used.
5	Waiting for Trigger (TRIG). Indicates whether the instrument is waiting for a trigger. This bit is set when CH 1 or another channel is waiting for a trigger. It is reset when the waiting-for-trigger status is canceled.
4– 1	Not used.
0	Calibration (CAL): Indicates whether the instrument is being calibrated. This bit is set when calibration is in progress and is reset when calibration ends.

**Operation Event Register (OEVR)**

In this instrument, this register has the same content as the Operation Condition Register (OCR), described above.

**Questionable Condition Register (QCR)**

The QCR is made up of 16 bits, which note the occurrence of only one type of event, as explained below.

15	14	13	12	11	10	9	8	7	6	5 FREQ	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	-----------	---	---	---	---	---

**Figure 3–5: The Questionable Condition Register (QCR)**

**Table 3–4: QCR bit functions**

Bit	Function
15 – 6	Not used. Must be set to zero for the waveform generator operation.
5	Frequency (FREQ). Indicates whether frequency accuracy of the signal is of questionable quality.
4 – 0	Not used. Must be set to zero for the waveform generator operation.

**Questionable Event Register (QEVR)**

This register holds the same content as the Questionable Condition Register (QCR).

**Enable Registers**

There are four types of enable registers:

- Event Status Enable Register (ESER)
- Service Request Enable Register (SRER)
- Operation Enable Register (OENR)
- Questionable Enable Register (QENR)

Each bit in the enable registers corresponds to a bit in the controlling status register. By setting and resetting the bits in the enable register, you can determine whether or not events that occur will be registered to the status register and queue.

**Event Status Enable Register (ESER)**

The ESER is made up of bits defined exactly the same as bits 0 through 7 in the SESR register (see Figure 3–6). You can use this register to designate whether or not the SBR ESB bit should be set when an event has occurred, and to determine if the corresponding SESR bit is set.

To set the SBR ESB bit (when the SESR bit has been set), set the ESER bit corresponding to that event. To prevent the ESB bit from being set, reset the ESER bit corresponding to that event.

Use the \*ESE command to set the bits of the ESER. Use the \*ESE? query to read the contents of the ESER.

7	6	5	4	3	2	1	0
PON	—	CME	EXE	DDE	QYE	—	OPC

**Figure 3–6: The Event Status Enable Register (ESER)**

**Service Request Enable Register (SRER)**

The SRER is made up of bits defined exactly the same as bits 0 through 7 in the SBR (see Figure 3–7). You can use this register to define which events will generate service requests.

The SRER bit 6 cannot be set. Also, the RQS is not maskable.

The generation of a service request with the GPIB interface involves changing the SRQ line to LOW, and making a service request to the controller. The result is that a status byte for which an RQS has been set is returned in response to serial polling by the controller.

Use the \*SRE command to set the bits of the SRER. Use the \*SRE? query to read the contents of the SRER. Bit 6 must be set to 0.

7	6	5	4	3	2	1	0
OSS	—	ESB	MAV	QSS	EAV	—	—

**Figure 3-7: The Service Request Enable Register (SRER)**

**Operation Enable Register (OENR)**

The OENR is made up of bits that are defined exactly the same as bits 0 through 15 in the OEVR register (see Figure 3-8). The operator uses this register to define whether or not the OSS bit in the SBR is set when an event occurs and the corresponding OEVR bit is set.

Use the STATUS:OPERation:ENABle command to set the bits in the OENR. Use the STATUS:OPERation:ENABle? query to read the contents of the OENR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										TRIG					CAL

**Figure 3-8: The Operation Enable Register (OENR)**

**Questionable Enable Register (QENR)**

The QENR is made up of bits that are defined exactly the same as bits 0 through 15 in the QEVR register (see Figure 3-9). You can use this register to define whether the QSS bit in the SBR is set when an event occurs and the corresponding QEVR bit is set.

Use the STATUS:QUESTionable:ENABle command to set the bits in the QENR. Use the STATUS:QUESTionable:ENABle? query to read the contents of the QENR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										FREQ					

**Figure 3-9: The Questionable Enable Register (QENR)**

## Queues

There are two types of queues in the status reporting system: output queues and error/event queues.

### Output Queue

The output queue is a FIFO (first-in, first-out) queue that holds response messages to queries awaiting retrieval. When there are messages in the queue, the SBR MAV bit is set.

The output queue is emptied each time a command or query is received, so the controller must read the output queue before the next command or query is issued. If this is not done, an error occurs and the output queue is emptied; however, the operation proceeds even if an error occurs.

### Error/Event Queue

The event queue is a FIFO queue, which stores events as they occur in the instrument. If more than 64 events are stored, the 64th event is replaced with event code -350 (“Queue Overflow”).

The oldest error code and text are retrieved by using one of the following queries:

- `SYSTem:ERRor[:NEXT]?`

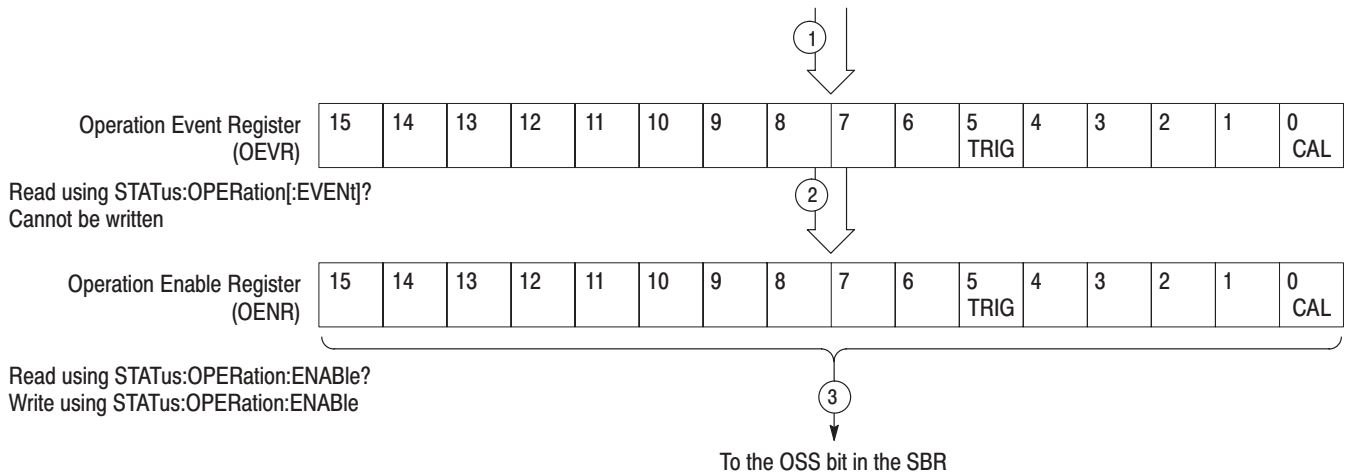
First, issue the `*ESR?` query to read the contents of the SESR. The contents of the SESR are cleared after they are read. If an SESR bit is set, events are stacked in the Error/Event Queue. Retrieve the event code with the following command sequence:

```
*ESR?  
SYSTem:ERRor[:NEXT]?
```

If you omit the `*ESR?` query, the SESR bit will remain set, even if the event disappears from the Error/Event Queue.

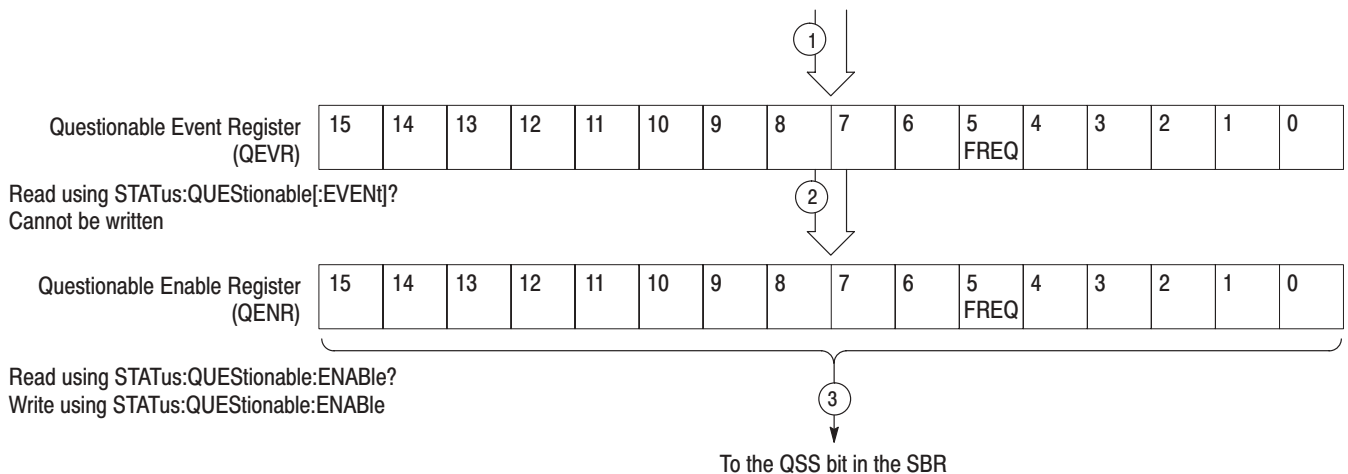
## Status and Event Processing Sequence

**Operation Status Block** As illustrated in Figure 3–10 below, a signal is sent to the OEVR (1) when an event occurs. If the corresponding bit in the OENR is also enabled (2), the OSS bit in the SBR is set to one (3). [See Figure 3–12 on page 3–12].



**Figure 3–10: Status and Event processing sequence — Operation status block**

**Questionable Status Block** As illustrated in Figure 3–11, when an event occurs, a signal is sent to the QEVR (1). If the corresponding bit in the QENR is also enabled (2), the QSS bit in the SBR is set to one (3). [See Figure 3–12 on on page 3–12].



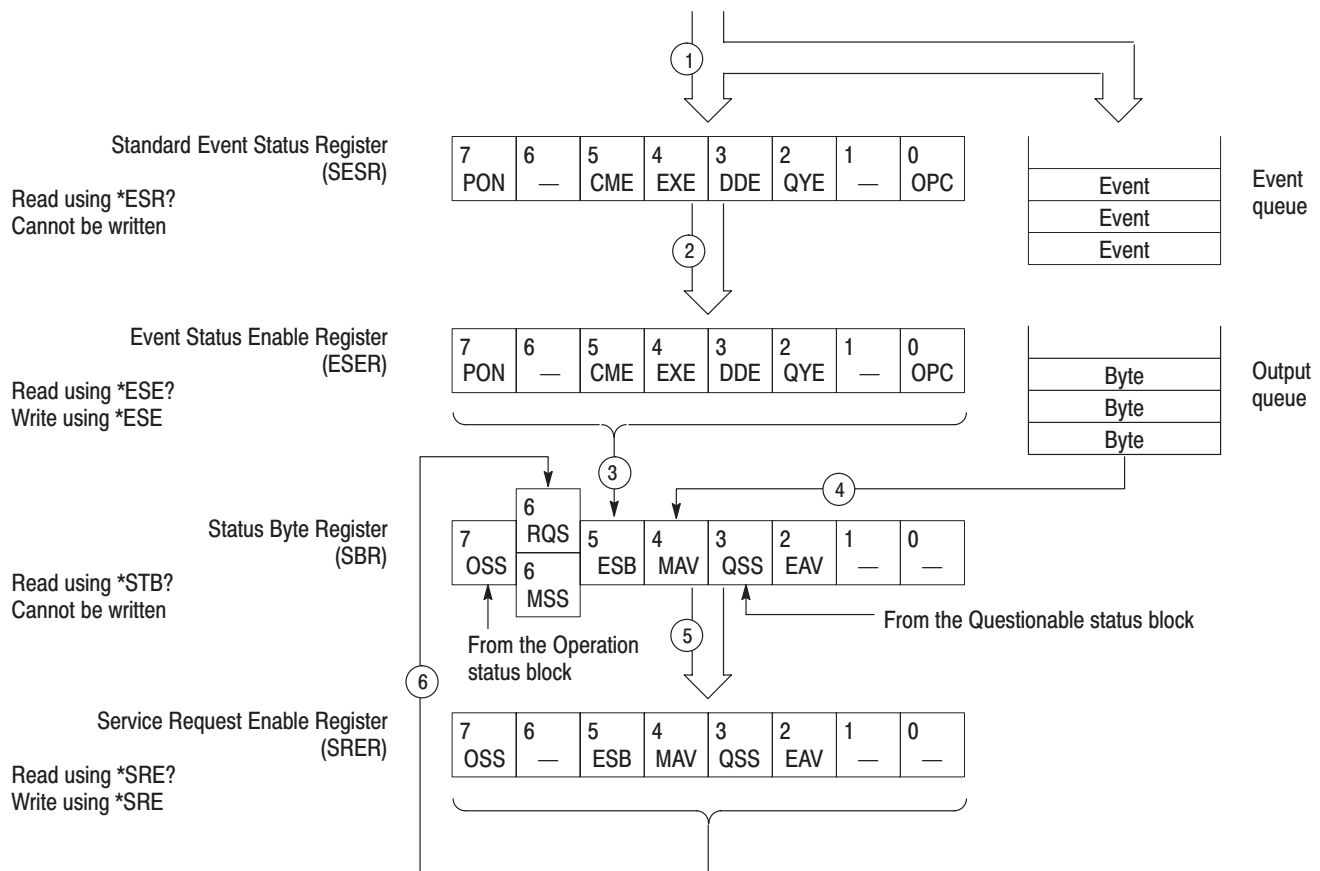
**Figure 3–11: Status and Event processing sequence — Questionable status block**

**Standard/Event Status Block**

As illustrated in Figure 3–12, when an event occurs, a signal is sent to the SESR and the event is recorded in the Event Queue (1). If the corresponding bit in the ESER is also enabled (2), the ESB bit in the SBR is set to one (3).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (4).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (5), the MSS bit in the SBR is set to one and a service request is generated (6).



**Figure 3–12: Status and Event processing sequence — Standard/Event status block**



## I/O Status and Event Screen

Figure 3–13 shows the contents of the GPIB status and event reporting system displayed on the status and event screen. Use this procedure to display the screen:

1. Press the **UTILITY** menu button on the front panel. The UTILITY menu appears on the screen.
2. Press the **Status** bottom menu button to display the Status submenu.
3. Press the **SCPI registers** side menu button to display the status and event screen.

The status and event screen displays the registers: SESR, ESER, SBR, SRER, OEVR, and QEVR. Each of these registers is displayed with the decimal equivalent of its contents shown in brackets. All events currently in the queue are listed in the Event Queue area of the display.

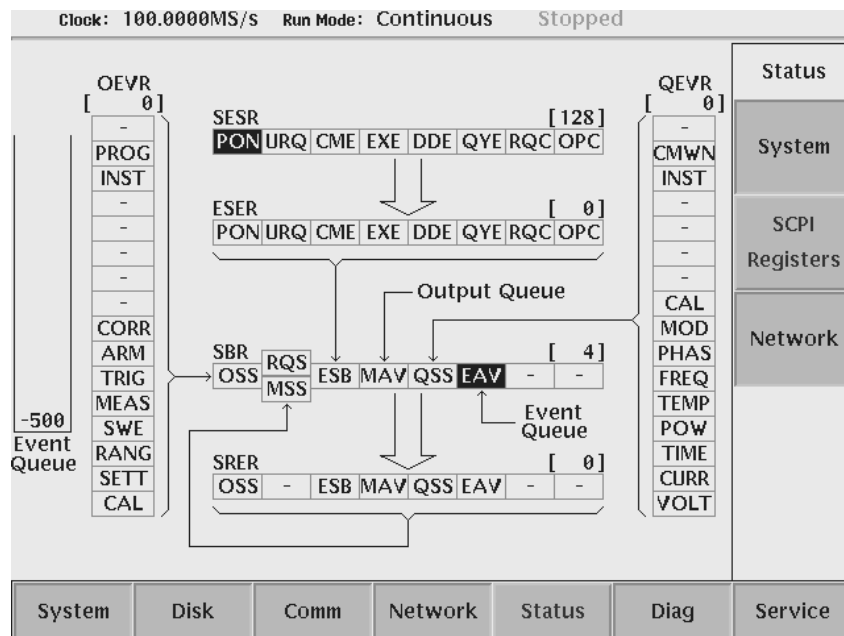


Figure 3–13: Status and Event screen

## Synchronizing Execution

All commands used in the waveform generator are designed to be executed in the order in which they are sent from the external controller. The following synchronization commands are included to ensure compliance with the SCPI standard.

- \*WAI
- \*OPC
- \*OPC?

## Messages

Tables 3–6 through 3–15 show the codes and messages used in the status and event reporting system.

Event codes and messages can be obtained by using the queries `SYSTEM:ERROR[:NEXT]?`. Responses are returned in the following format:

`<event code>,"<event message>"`

# Messages and Codes

Error and event codes with negative values are SCPI standard codes. Error and event codes with positive values are unique to the waveform generator series number.

Table 3–5 lists event code definitions. When an error occurs, you can find its error class by checking for its the code range in Tables 3–6 through 3–15. Events in these tables are organized by event class.

**Table 3–5: Definition of event codes**

<b>Event class</b>	<b>Code range</b>	<b>Description</b>
No error	0	No event or status
Command errors	–100 to –199	Command syntax errors
Execution errors	–200 to –299	Command execution errors
Device-specific errors	–300 to –399	Internal device errors
Query errors	–400 to –499	System event and query errors
Power-on events	–500 to –599	Power-on events
User request events	–600 to –699	User request events
Request control events	–700 to –799	Request control events
Operation complete events	–800 to –899	Operation complete events
Extended device-specific errors	1 to 32767	Device dependent device errors
Reserved	other than above	not used

## Command Errors

Command errors are returned when there is a syntax error in the command.

**Table 3-6: Command errors**

<b>Error code</b>	<b>Error message</b>
-100	Command error
-101	Invalid character
-102	Syntax error
-103	Invalid separator
-104	Data type error
-105	GET not allowed
-108	Parameter not allowed
-109	Missing parameter
-110	Command header error
-111	Header separator error
-112	Program mnemonic too long
-113	Undefined header
-114	Header suffix out of range
-115	Unexpected number of parameters
-120	Numeric data error
-121	Invalid character in number
-123	Exponent too large
-124	Too many digits
-128	Numeric data not allowed

**Table 3-6: Command errors (Cont.)**

<b>Error code</b>	<b>Error message</b>
-130	Suffix error
-131	Invalid suffix
-134	Suffix too long
-138	Suffix not allowed
-140	Character data error
-141	Invalid character data
-144	Character data too long
-148	Character data not allowed
-150	String data error
-151	Invalid string data
-158	String data not allowed
-160	Block data error
-161	Invalid block data
-168	Block data not allowed
-170	Expression error
-171	Invalid expression
-178	Expression data not allowed
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error

## Execution Errors

These error codes are returned when an error is detected during command execution.

**Table 3-7: Execution errors**

<b>Error code</b>	<b>Error message</b>
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to RTL
-203	Command protected
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-224	Illegal parameter value
-225	Out of memory
-226	Lists not same length
-230	Data corrupt or stale
-231	Data questionable
-232	Invalid format
-233	Invalid version
-240	Hardware error
-241	Hardware missing
-250	Mass storage error
-251	Missing mass storage
-252	Missing media

**Table 3-7: Execution errors (Cont.)**

<b>Error code</b>	<b>Error message</b>
-253	Corrupt media
-254	Media full
-255	Directory full
-256	File name not found
-257	File name error
-258	Media protected
-260	Expression error
-261	Math error in expression
-270	Macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label
-274	Macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error
-290	Memory use error
-291	Out of memory
-292	Referenced name does not exist
-293	Referenced name already exists
-294	Incompatible type

## Device Specific Errors

These error codes are returned when an internal instrument error is detected. This type of error can indicate a hardware problem.

**Table 3-8: Device specific errors**

<b>Error code</b>	<b>Error message</b>
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory lost
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-320	Storage fault
-321	Out of memory
-330	Self-test failed
-340	Calibration failed
-350	Queue overflow
-360	Communication error
-361	Parity error in program message
-362	Framing error in program message
-363	Input buffer overrun
-365	Time out error



## Query Errors

These error codes are returned in response to an unanswered query.

**Table 3-9: Query errors**

Error code	Error message
-400	query error
-410	query INTERRUPTED
-420	query UNTERMINATED
-430	query DEADLOCKED
-440	query UNTERMINATED after indefinite response

## Power-On Events

These events occur when the instrument detects an off to on transition in its power supply.

**Table 3-10: Power-on events**

Event code	Event message
-500	Power on

## User Request Events

These events are unused in AWG.

**Table 3-11: User request events**

Event code	Event message
-600	User request

## Request Control Events

This event is unused in AWG.

**Table 3-12: Request control events**

Event code	Event message
-700	Request control

## Operation Complete Events

This event occurs when the instrument's synchronization protocol, having been enabled by an \*OPC command, completes all selected pending operations.

**Table 3-13: Operation complete events**

Event code	Event message
-800	Operation complete

## Device Errors

The error codes in Table 3–14 are unique to AWG400 series. For AWG500/600 series error codes, see Table 3–15 on 3–30.

**Table 3–14: Device errors (AWG400 series)**

Error code	Error message
1101	CH1 Internal Offset calibration failure
1102	CH2 Internal Offset calibration failure
1103	CH3 Internal Offset calibration failure
1104	CH1 Internal Offset calibration failure
1105	CH2 Internal Offset calibration failure
1106	CH3 Internal Offset calibration failure
1201	CH1 Output Offset calibration failure
1202	CH2 Output Offset calibration failure
1203	CH3 Output Offset calibration failure
1204	CH1 Output Offset calibration failure
1205	CH2 Output Offset calibration failure
1206	CH3 Output Offset calibration failure
1301	CH1 Gain calibration failure
1302	CH2 Gain calibration failure
1303	CH3 Gain calibration failure
1304	CH1 Gain calibration failure
1305	CH2 Gain calibration failure
1306	CH3 Gain calibration failure
1401	CH1 Direct Output Gain calibration failure
1402	CH2 Direct Output Gain calibration failure
1403	CH3 Direct Output Gain calibration failure
1404	CH1 Direct Output Gain calibration failure
1405	CH2 Direct Output Gain calibration failure
1406	CH3 Direct Output Gain calibration failure
1501	CH1 Direct Output Gain calibration failure
1502	CH2 Direct Output Gain calibration failure
1503	CH3 Direct Output Gain calibration failure
1504	CH1 Direct Output Gain calibration failure
1505	CH2 Direct Output Gain calibration failure
1506	CH3 Direct Output Gain calibration failure
1601	CH1 Attenuator calibration failure
1602	CH2 Attenuator calibration failure

**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
1603	CH3 Attenuator calibration failure
1604	CH1 Attenuator calibration failure
1605	CH2 Attenuator calibration failure
1606	CH3 Attenuator calibration failure
1611	CH1 x5dB Attenuator calibration failure
1612	CH2 x5dB Attenuator calibration failure
1613	CH3 x5dB Attenuator calibration failure
1614	CH1 x5dB Attenuator calibration failure
1615	CH2 x5dB Attenuator calibration failure
1616	CH3 x5dB Attenuator calibration failure
1621	CH1 x10dB 1 Attenuator calibration failure
1622	CH2 x10dB 1 Attenuator calibration failure
1623	CH3 x10dB 1 Attenuator calibration failure
1624	CH1 x10dB 1 Attenuator calibration failure
1625	CH2 x10dB 1 Attenuator calibration failure
1626	CH3 x10dB 1 Attenuator calibration failure
1631	CH1 x10dB 2 Attenuator calibration failure
1632	CH2 x10dB 2 Attenuator calibration failure
1633	CH3 x10dB 2 Attenuator calibration failure
1634	CH1 x10dB 2 Attenuator calibration failure
1635	CH2 x10dB 2 Attenuator calibration failure
1636	CH3 x10dB 2 Attenuator calibration failure
1641	CH1 x20dB Attenuator calibration failure
1642	CH2 x20dB Attenuator calibration failure
1643	CH3 x20dB Attenuator calibration failure
1644	CH1 x20dB Attenuator calibration failure
1645	CH2 x20dB Attenuator calibration failure
1646	CH3 x20dB Attenuator calibration failure
1701	CH1 Filter calibration failure
1702	CH2 Filter calibration failure
1703	CH3 Filter calibration failure
1704	CH1 Filter calibration failure
1705	CH2 Filter calibration failure
1706	CH3 Filter calibration failure
1711	CH1 1MHz Filter calibration failure
1712	CH2 1MHz Filter calibration failure
1713	CH3 1MHz Filter calibration failure

**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
1714	CH1 1MHz Filter calibration failure
1715	CH2 1MHz Filter calibration failure
1716	CH3 1MHz Filter calibration failure
1721	CH1 5MHz Filter calibration failure
1722	CH2 5MHz Filter calibration failure
1723	CH3 5MHz Filter calibration failure
1724	CH1 5MHz Filter calibration failure
1725	CH2 5MHz Filter calibration failure
1726	CH3 5MHz Filter calibration failure
1731	CH1 20MHz Filter calibration failure
1732	CH2 20MHz Filter calibration failure
1733	CH3 20MHz Filter calibration failure
1734	CH1 20MHz Filter calibration failure
1735	CH2 20MHz Filter calibration failure
1736	CH3 20MHz Filter calibration failure
1741	CH1 50MHz Filter calibration failure
1742	CH2 50MHz Filter calibration failure
1743	CH3 50MHz Filter calibration failure
1744	CH1 50MHz Filter calibration failure
1745	CH2 50MHz Filter calibration failure
1746	CH3 50MHz Filter calibration failure
2100	System failure
2101	Real-time clock power
2102	Configuration record and checksum status
2103	Incorrect configuration
2104	Memory size miscompare
2105	Fixed-disk drive initialization status
2106	Time status
2110	Front panel failure
2111	Front panel configuration
2112	Front panel communication
2113	Front panel RAM
2114	Front panel ROM
2115	Front panel A/D
2116	Front panel timer
2700	Calibration data failure
2701	Calibration data not found

**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
2702	Calibration data checksum
2703	Calibration data invalid
3000	Run mode failure
3100	Run mode control1 register failure
3101 to 3132	Run mode control1 register bit0 to bit31
4000	Clock failure
4100	PLL lock/unlock failure
5000	Sequence memory failure
5100	Sequence memory data bus failure
5101 to 5132	Sequence memory data bus bit0 to bit31
5200	Sequence memory address bus failure
5201 to 5224	Sequence memory address bus bit0 to bit23
5300	Sequence memory chip cell failure
5301 to 5302	Sequence memory chip 0 to chip 1
5350	Sequence memory chip select failure
5351 to 5352	Sequence memory chip select 0 to select 1
5600	Event table memory data bus failure
5601 to 5632	Event table memory data bus bit0 to bit31
5700	Event table memory address bus failure
5701 to 5704	Event table memory address bus bit0 to bit3
5800	Event table memory chip cell failure
5801 to 5802	Event table memory chip 0 to chip 1
6000	Waveform memory failure
6100	CH1 Waveform memory data bus failure
6101 to 6132	CH1 Waveform memory data bus bit0 to bit31
6200	CH1 Waveform memory address bus failure
6201 to 6224	CH1 Waveform memory address bus bit0 to bit23
6300	CH1 Waveform memory chip cell failure
6301 to 6332	CH1 Waveform memory chip 0 to chip 31
6350	CH1 Waveform memory chip select failure
6351 to 6382	CH1 Waveform memory chip select 0 to select 31
6400	CH2 Waveform memory data bus failure
6401 to 6432	CH2 Waveform memory data bus bit0 to bit31
6500	CH2 Waveform memory address bus failure
6501 to 6524	CH2 Waveform memory address bus bit0 to bit23
6600	CH2 Waveform memory chip cell failure
6601 to 6632	CH2 Waveform memory chip 0 to chip 31

**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
6650	CH2 Waveform memory chip select failure
6651 to 6682	CH2 Waveform memory chip select 0 to select 31
6700	CH3 Waveform memory data bus failure
6701 to 6732	CH3 Waveform memory data bus bit0 to bit31
6800	CH3 Waveform memory address bus failure
6801 to 6824	CH3 Waveform memory address bus bit0 to bit23
6900	CH3 Waveform memory chip cell failure
6901 to 6932	CH3 Waveform memory chip 0 to chip 31
6950	CH3 Waveform memory chip select failure
6951 to 6982	CH3 Waveform memory chip select 0 to select 31
7000	Output failure
7100	Internal offset failure
7101	CH1 internal offset
7102	CH2 internal offset
7103	CH3 internal offset
7104	$\overline{\text{CH1}}$ internal offset
7105	$\overline{\text{CH2}}$ internal offset
7106	$\overline{\text{CH3}}$ internal offset
7200	Output offset failure
7201	CH1 output offset
7202	CH2 output offset
7203	CH3 output offset
7204	$\overline{\text{CH1}}$ output offset
7205	$\overline{\text{CH2}}$ output offset
7206	$\overline{\text{CH3}}$ output offset
7300	Arb gain failure
7301	CH1 Arb gain
7302	CH2 Arb gain
7303	CH3 Arb gain
7304	$\overline{\text{CH1}}$ Arb gain
7305	$\overline{\text{CH2}}$ Arb gain
7306	$\overline{\text{CH3}}$ Arb gain
7400	Direct gain failure
7401	CH1 Direct gain
7402	CH2 Direct gain
7403	CH3 Direct gain
7404	$\overline{\text{CH1}}$ Direct gain

**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
7405	CH2 Direct gain
7406	CH3 Direct gain
7510	5dB attenuator failure
7511	CH1 5dB attenuator
7512	CH2 5dB attenuator
7513	CH3 5dB attenuator
7514	CH1 5dB attenuator
7515	CH2 5dB attenuator
7516	CH3 5dB attenuator
7520	10dB attenuator 1 failure
7521	CH1 10dB 1 attenuator
7522	CH2 10dB 1 attenuator
7523	CH3 10dB 1 attenuator
7524	CH1 10dB 1 attenuator
7525	CH2 10dB 1 attenuator
7526	CH3 10dB 1 attenuator
7530	10dB attenuator 2 failure
7531	CH1 10dB 2 attenuator
7532	CH2 10dB 2 attenuator
7533	CH3 10dB 2 attenuator
7534	CH1 10dB 2 attenuator
7535	CH2 10dB 2 attenuator
7536	CH3 10dB 2 attenuator
7540	20dB attenuator failure
7541	CH1 20dB attenuator
7542	CH2 20dB attenuator
7543	CH3 20dB attenuator
7544	CH1 20dB attenuator
7545	CH2 20dB attenuator
7546	CH3 20dB attenuator
7610	1MHz filter failure
7611	CH1 1MHz filter
7612	CH2 1MHz filter
7613	CH3 1MHz filter
7614	CH1 1MHz filter
7615	CH2 1MHz filter
7616	CH3 1MHz filter



**Table 3-14: Device errors (AWG400 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
7620	5MHz filter failure
7621	CH1 5MHz filter
7622	CH2 5MHz filter
7623	CH3 5MHz filter
7624	$\overline{\text{CH1}}$ 5MHz filter
7625	$\overline{\text{CH2}}$ 5MHz filter
7626	$\overline{\text{CH3}}$ 5MHz filter
7630	20MHz filter failure
7631	CH1 20MHz filter
7632	CH2 20MHz filter
7633	CH3 20MHz filter
7634	$\overline{\text{CH1}}$ 20MHz filter
7635	$\overline{\text{CH2}}$ 20MHz filter
7636	$\overline{\text{CH3}}$ 20MHz filter
7640	50MHz filter failure
7641	CH1 50MHz filter
7642	CH2 50MHz filter
7643	CH3 50MHz filter
7644	$\overline{\text{CH1}}$ 50MHz filter
7645	$\overline{\text{CH2}}$ 50MHz filter
7646	$\overline{\text{CH3}}$ 50MHz filter
9111	Waveform/Sequence load error: waveform memory full
9112	Waveform/Sequence load error: invalid waveform length
9113	Waveform/Sequence load error: waveform length too short
9114	Waveform/Sequence load error: waveform length changed
9121	Sequence load error: missing file name in sequence
9122	Sequence load error: too many nesting levels
9123	Sequence load error: infinite loop in sub-sequence
9124	Sequence load error: infinite sub-sequence loop
9125	Sequence load error: max sequence elements exceeded
9126	Sequence load error: invalid jump address
9127	Sequence load error: sequence memory full
9128	Sequence load error: infinite loop and Goto One not allowed
9151	Waveform load warning: output disabled in some channels
9152	Waveform/Sequence output warning: output disabled

Table 3–15 lists error codes for the AWG500/600 series.

**Table 3–15: Device errors (AWG500/600 series)**

Error code	Error message
1101	CH1 internal offset
1102	CH1 output offset
1103	CH1 gain
1104	CH1 gain difference
1105	CH1 direct output gain
1111	CH1 x3dB attenuator
1112	CH1 x6dB attenuator
1113	CH1 x12dB attenuator
1114	CH1 x20dB attenuator
1115	CH1 x5dB 1 attenuator
1116	CH1 x5dB 2 attenuator
1117	CH1 x10dB 2 attenuator
1121	CH1 10MHz filter
1122	CH1 20MHz filter
1123	CH1 50MHz filter
1124	CH1 100MHz filter
1125	CH1 200MHz filter
1201	CH2 or $\overline{\text{CH1}}$ internal offset
1202	CH2 or $\overline{\text{CH1}}$ output offset
1203	CH2 or $\overline{\text{CH1}}$ gain
1204	CH2 or $\overline{\text{CH1}}$ gain difference
1205	CH2 or $\overline{\text{CH1}}$ direct output gain
1211	CH2 or $\overline{\text{CH1}}$ x3dB attenuator
1212	CH2 or $\overline{\text{CH1}}$ x6dB attenuator
1213	CH2 or $\overline{\text{CH1}}$ x12dB attenuator
1214	CH2 or $\overline{\text{CH1}}$ x20dB attenuator
1215	CH2 or $\overline{\text{CH1}}$ x5dB 1 attenuator
1216	CH2 or $\overline{\text{CH1}}$ x5dB 2 attenuator
1217	CH2 or $\overline{\text{CH1}}$ x10dB 2 attenuator
1221	CH2 or $\overline{\text{CH1}}$ 10MHz filter
1222	CH2 or $\overline{\text{CH1}}$ 20MHz filter

**Table 3-15: Device errors (AWG500/600 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
1223	CH2 or CH1 50MHz filter
1224	CH2 or CH1 100MHz filter
1225	CH2 or CH1 200MHz filter
2100	System failure
2101	Real-time clock power
2102	Configuration record and checksum status
2103	Incorrect configuration
2104	Memory size miscompare
2105	Fixed-disk drive initialization status
2106	Time status
2110	Front panel failure
2111	Front panel configuration
2112	Front panel communication
2113	Front panel RAM
2114	Front panel ROM
2115	Front panel A/D
2116	Front panel timer
2301	A30 board failure
2401	Clock delay data not found
2402	Clock delay data checksum
2700	Calibration data failure
2701	Calibration data not found
2702	Calibration data checksum
2703	Calibration data invalid
3100	Control1 register failure
3101 to 3104	Control1 register bit0 to bit3
3200	Event table data bus failure
3201 to 3216	Event table data bus bit0 to bit15
3250	Event table address bus failure
3251 to 3254	Event table address bus bit0 to bit3
3300	Event table memory chip select failure
3301	Event table memory chip select 0
3302	Event table memory chip select 1

**Table 3-15: Device errors (AWG500/600 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
3350	Event table memory chip cell failure
3351	Event table memory chip 0
3352	Event table memory chip 1
4100	A40 board failure
4101	PLL 500MHz locked
4102	PLL 600MHz unlocked
4103	PLL 1350MHz unlocked
4104	PLL 1500MHz locked
4105	Low band VCO PLL error
4106	High band VCO PLL error
5100	Sequence memory data bus failure
5101 to 5116	Sequence memory data bus bit0 to bit15
5117 to 5132	Sequence memory high data bus bit0 to bit15
5133 to 5148	Sequence memory opcode data bus bit0 to bit15
5150	Sequence memory address bus failure
5151 to 5174	Sequence memory address bus bit0 to bit23
5200	Sequence memory chip select failure
5201 to 5206	Sequence memory chip select 0 to select 5
5250	Sequence memory chip cell failure
5251 to 5256	Sequence memory chip 0 to chip 5
5300	CH1 Waveform memory data bus failure
5301 to 5316	CH1 Waveform memory data bus bit0 to bit15
5330	CH1 Waveform memory module data bus failure
5331 to 5340	CH1 Waveform memory module data bus module 0 to module 9
5350	CH1 Waveform memory address bus failure
5351 to 5374	CH1 Waveform memory address bus bit0 to bit23
5400	CH1 Waveform memory chip select failure
5401 to 5449	CH1 Waveform memory chip select 0 to select 48
5500	CH1 Waveform memory chip cell failure
5501 to 5549	CH1 Waveform memory chip 0 to chip 48
5550 to 5580	CH1 Waveform memory chip 49 to chip 79
5600	CH2 or $\overline{\text{CH1}}$ Waveform memory data bus failure
5601 to 5616	CH2 or $\overline{\text{CH1}}$ Waveform memory data bus bit0 to bit15

**Table 3–15: Device errors (AWG500/600 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
5650	CH2 or $\overline{\text{CH1}}$ Waveform memory address bus failure
5651 to 5674	CH2 or $\overline{\text{CH1}}$ Waveform memory address bus bit0 to bit23
5700	CH2 or $\overline{\text{CH1}}$ Waveform memory chip select failure
5701 to 5748	CH2 or $\overline{\text{CH1}}$ Waveform memory chip select 0 to select 47
5800	CH2 or $\overline{\text{CH1}}$ Waveform memory chip cell failure
5801 to 5848	CH2 or $\overline{\text{CH1}}$ Waveform memory chip 0 to chip 47
5900	CH1 Arb D/A failure
5901 to 5912	CH1 Arb D/A data bit0 to bit11
5950	CH2 Arb D/A failure
5951 to 5962	CH2 Arb D/A data bit0 to bit11
7110	CH1 output offset failure
7111	CH1 output offset
7120	CH1 internal offset failure
7121	CH1 internal offset
7130	CH1 Arb gain failure
7131	CH1 Arb gain
7140	CH1 attenuator failure
7141	CH1 3dB attenuator
7142	CH1 6dB attenuator
7143	CH1 12dB attenuator
7144	CH1 20dB attenuator
7145	CH1 5dB 1 attenuator
7146	CH1 5dB 2 attenuator
7147	CH1 10dB attenuator
7150	CH1 filter failure
7151	CH1 10MHz filter
7152	CH1 20MHz filter
7153	CH1 50MHz filter
7154	CH1 100MHz filter
7155	CH1 200MHz filter
7170	CH1 output key failure
7171	CH1 output key
7210	CH2 or $\overline{\text{CH1}}$ output offset failure

**Table 3-15: Device errors (AWG500/600 series) (Cont.)**

<b>Error code</b>	<b>Error message</b>
7211	CH2 or $\overline{\text{CH1}}$ output offset
7220	CH2 or $\overline{\text{CH1}}$ internal offset failure
7221	CH2 or $\overline{\text{CH1}}$ internal offset
7230	CH2 or $\overline{\text{CH1}}$ Arb gain failure
7231	CH2 or $\overline{\text{CH1}}$ Arb gain
7240	CH2 or $\overline{\text{CH1}}$ attenuator failure
7241	CH2 or $\overline{\text{CH1}}$ 3dB attenuator
7242	CH2 or $\overline{\text{CH1}}$ 6dB attenuator
7243	CH2 or $\overline{\text{CH1}}$ 12dB attenuator
7244	CH2 or $\overline{\text{CH1}}$ 20dB attenuator
7250	CH2 or $\overline{\text{CH1}}$ filter failure
7251	CH2 or $\overline{\text{CH1}}$ 10MHz filter
7252	CH2 or $\overline{\text{CH1}}$ 20MHz filter
7253	CH2 or $\overline{\text{CH1}}$ 50MHz filter
7254	CH2 or $\overline{\text{CH1}}$ 100MHz filter
7270	CH2 or $\overline{\text{CH1}}$ output key failure
7271	CH2 or $\overline{\text{CH1}}$ output key
9111	Waveform/Sequence load error: waveform memory full
9112	Waveform/Sequence load error: invalid waveform length
9113	Waveform/Sequence load error: waveform length too short
9114	Waveform/Sequence load error: waveform length changed
9121	Sequence load error: missing file name in sequence
9122	Sequence load error: too many nesting levels
9123	Sequence load error: infinite loop in subsequence
9124	Sequence load error: infinite subsequence loop
9125	Sequence load error: max sequence elements exceeded
9126	Sequence load error: invalid jump address
9127	Sequence load error: sequence memory full
9128	Sequence load error: infinite loop and Goto One not allowed
9151	Waveform load warning: output disabled in some channels
9152	Waveform/Sequence output warning: output disabled



# Examples



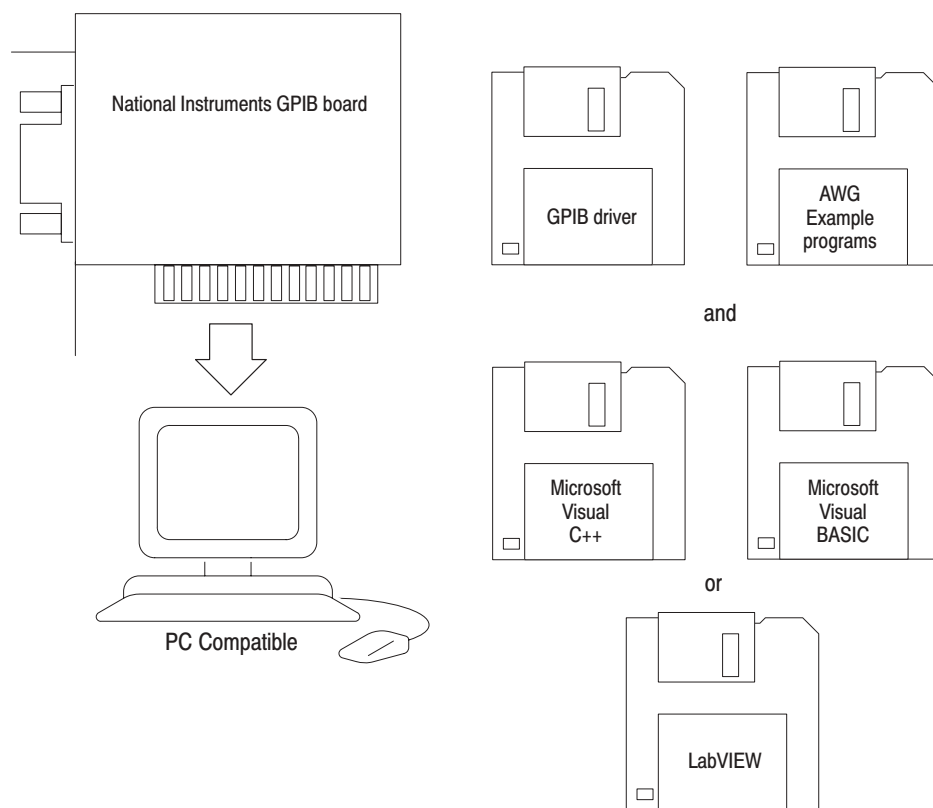


# Programming Examples

The floppy disk supplied with the waveform generator contains programming samples to use the GPIB and Ethernet interfaces. These programs are written in Microsoft Visual C++ and Visual BASIC.

Figure 4–1 displays what you need to run the GPIB example programs. GPIB programs run on a PC-compatible system. To use the GPIB interface, your PC-compatible system must be equipped with a National Instruments GPIB board and associated drivers. GPIB programs are also compatible with National Instruments LabVIEW.

The diskette also contains the file *README.TXT*. Refer to the file for details about how to run the programs.



**Figure 4–1: Equipment needed to run the GPIB example programs**





# Appendices



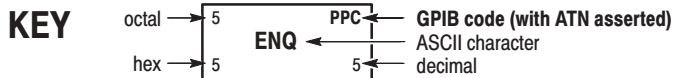
# Appendix A: Character Charts

Table A-1: The AWG character set

	0	1	2	3	4	5	6	7
<b>0</b>	<b>NUL</b> 0	16	<b>space</b> 32	<b>0</b> 48	<b>@</b> 64	<b>P</b> 80	<b>'</b> 96	<b>p</b> 112
<b>1</b>	1	17	<b>!</b> 33	<b>1</b> 49	<b>A</b> 65	<b>Q</b> 81	<b>a</b> 97	<b>q</b> 113
<b>2</b>	2	18	<b>”</b> 34	<b>2</b> 50	<b>B</b> 66	<b>R</b> 82	<b>b</b> 98	<b>r</b> 114
<b>3</b>	3	19	<b>#</b> 35	<b>3</b> 51	<b>C</b> 67	<b>S</b> 83	<b>c</b> 99	<b>s</b> 115
<b>4</b>	4	20	<b>\$</b> 36	<b>4</b> 52	<b>D</b> 68	<b>T</b> 84	<b>d</b> 100	<b>t</b> 116
<b>5</b>	5	21	<b>%</b> 37	<b>5</b> 53	<b>E</b> 69	<b>U</b> 85	<b>e</b> 101	<b>u</b> 117
<b>6</b>	6	22	<b>&amp;</b> 38	<b>6</b> 54	<b>F</b> 70	<b>V</b> 86	<b>f</b> 102	<b>v</b> 118
<b>7</b>	7	23	<b>,</b> 39	<b>7</b> 55	<b>G</b> 71	<b>W</b> 87	<b>g</b> 103	<b>w</b> 119
<b>8</b>	8	24	<b>(</b> 40	<b>8</b> 56	<b>H</b> 72	<b>X</b> 88	<b>h</b> 104	<b>x</b> 120
<b>9</b>	<b>HT</b> 9	25	<b>)</b> 41	<b>9</b> 57	<b>I</b> 73	<b>Y</b> 89	<b>i</b> 105	<b>y</b> 121
<b>A</b>	<b>LF</b> 10	26	<b>*</b> 42	<b>:</b> 58	<b>J</b> 74	<b>Z</b> 90	<b>j</b> 106	<b>z</b> 122
<b>B</b>	11	<b>ESC</b> 27	<b>+</b> 43	<b>;</b> 59	<b>K</b> 75	<b>[</b> 91	<b>k</b> 107	<b>{</b> 123
<b>C</b>	12	28	<b>,</b> 44	<b>&lt;</b> 60	<b>L</b> 76	<b>\</b> 92	<b>l</b> 108	<b> </b> 124
<b>D</b>	<b>CR</b> 13	29	<b>—</b> 45	<b>=</b> 61	<b>M</b> 77	<b>]</b> 93	<b>m</b> 109	<b>}</b> 125
<b>E</b>	14	30	<b>.</b> 46	<b>&gt;</b> 62	<b>N</b> 78	<b>^</b> 94	<b>n</b> 110	<b>~</b> 126
<b>F</b>	15	31	<b>/</b> 47	<b>?</b> 63	<b>O</b> 79	<b>_</b> 95	<b>o</b> 111	<b>rubout</b> 127

**Table A-2: ASCII & GPIB code chart**

B7 B6 BITS B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1				
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE						
0 0 0 0	0 0	NUL	20 10	DLE	40 20	SP	60 30	LA16	100 40	TA0	@	120 50	TA16	P	140 60	SA0	160 70	SA16	p
0 0 0 1	1 1	GTL SOH	21 11	LL0 DC1	41 21	!	61 31	LA17	101 41	TA1	A	121 51	TA17	Q	141 61	SA1	161 71	SA17	q
0 0 1 0	2 2	STX	22 12	DC2	42 22	"	62 32	LA18	102 42	TA2	B	122 52	TA18	R	142 62	SA2	162 72	SA18	r
0 0 1 1	3 3	ETX	23 13	DC3	43 23	#	63 33	LA19	103 43	TA3	C	123 53	TA19	S	143 63	SA3	163 73	SA19	s
0 1 0 0	4 4	SDC EOT	24 14	DCL DC4	44 24	\$	64 34	LA20	104 44	TA4	D	124 54	TA20	T	144 64	SA4	164 74	SA20	t
0 1 0 1	5 5	PPC ENQ	25 15	PPU NAK	45 25	%	65 35	LA21	105 45	TA5	E	125 55	TA21	U	145 65	SA5	165 75	SA21	u
0 1 1 0	6 6	ACK	26 16	SYN	46 26	&	66 36	LA22	106 46	TA6	F	126 56	TA22	V	146 66	SA6	166 76	SA22	v
0 1 1 1	7 7	BEL	27 17	ETB	47 27	'	67 37	LA23	107 47	TA7	G	127 57	TA23	W	147 67	SA7	167 77	SA23	w
1 0 0 0	10 8	GET BS	30 18	SPE CAN	50 28	(	70 38	LA24	110 48	TA8	H	130 58	TA24	X	150 68	SA8	170 78	SA24	x
1 0 0 1	11 9	TCT HT	31 19	SPD EM	51 29	)	71 39	LA25	111 49	TA9	I	131 59	TA25	Y	151 69	SA9	171 79	SA25	y
1 0 1 0	12 A	LF	32 1A	SUB	52 2A	*	72 3A	LA26	112 4A	TA10	J	132 5A	TA26	Z	152 6A	SA10	172 7A	SA26	z
1 0 1 1	13 B	VT	33 1B	ESC	53 2B	+	73 3B	LA27	113 4B	TA11	K	133 5B	TA27	[	153 6B	SA11	173 7B	SA27	{
1 1 0 0	14 C	FF	34 1C	FS	54 2C	,	74 3C	LA28	114 4C	TA12	L	134 5C	TA28	\	154 6C	SA12	174 7C	SA28	
1 1 0 1	15 D	CR	35 1D	GS	55 2D	-	75 3D	LA29	115 4D	TA13	M	135 5D	TA29	]	155 6D	SA13	175 7D	SA29	}
1 1 1 0	16 E	SO	36 1E	RS	56 2E	.	76 3E	LA30	116 4E	TA14	N	136 5E	TA30	^	156 6E	SA14	176 7E	SA30	~
1 1 1 1	17 F	SI	37 1F	US	57 2F	/	77 3F	UNL	117 4F	TA15	O	137 5F	UNT	-	157 6F	SA15	177 7F	SA30	RUBOUT (DEL)
		ADDRESSED COMMANDS		UNIVERSAL COMMANDS		LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS									



**Tektronix**  
REF: ANSI STD X3.4-1977  
IEEE STD 488.1-1987  
ISO STD 646-2973

# Appendix B: GPIB Interface Specification

This appendix lists and describes the GPIB functions and messages the waveform generator implements.

## Interface Functions

Table B–1 lists the GPIB interface functions this instrument implements. Each function is briefly described on the next page.

**Table B–1: GPIB interface function implementation**

<b>Interface function</b>	<b>Implemented subset</b>	<b>Capability</b>
Acceptor Handshake (AH)	AH1	Complete
Source Handshake (SH)	SH1	Complete
Talker (T)	T6	Basic Talker, Serial Poll Unaddress if my-listen-address (MLA) No Talk Only mode
Listener (L)	L4	Basic Listener Unaddress if my talk address (MTA) No Listen Only mode
Service Request (SR)	SR1	Complete
Remote/Local (RL)	RL1	Complete
Parallel Poll (PP)	PP0	None
Device Clear (DC)	DC1	Complete
Device Trigger (DT)	DT1	Complete
Controller (C)	C0	None
Electrical Interface	E2	Three-state driver

- Acceptor Handshake (AH). Enables a listening device to coordinate data reception. The AH function delays data transfer initiation or termination until the listening device is ready to receive the next data byte.
- Source Handshake (SH). Enables a talking device to support the coordination of data transfer. The SH function controls the initiation and termination of data byte transfers.
- Talker (T). Enables a device to send device-dependent data over the interface. This capability is available only when the device is addressed to talk, and uses a one-byte address.
- Listener (L). Enables a device to receive device-dependent data over the interface. This capability is available only when the device is addressed to listen, and uses a one-byte address.
- Service Request (SR). Enables a device to request service from the controller.
- Remote/Local (RL). Enables a device to select between one of two sources for waveform generator control. It determines whether input information is controlled from the front panel (local control) or by GPIB commands (remote control).
- Device Clear (DC). Enables a device to be cleared or initialized, either individually, or as part of a group of devices.
- Controller (C). Enables a device that has this capability to send its address, universal commands, and addressed commands to other devices over the interface.
- Electrical Interface (E). Identifies the electrical interface driver type. The notation E1 means the electrical interface uses open collector drivers, E2 means the electrical interface uses three-state drivers.



## Interface Messages

Table B-2 shows the standard interface messages the waveform generator supports. Brief function descriptions are provided on the next page.

**Table B-2: AWG standard interface message**

Message	GPIB
DCL	Yes
GET	Yes
GTL	Yes
LLO	Yes
PPC	No
PPD	No
PPE	No
PPU	No
SDC	Yes
SPD	Yes
SPE	Yes
TCT	No
UNL	Yes
UNT	Yes
Listen Addresses	Yes
Talk Addresses	Yes

- Device Clear (DCL). Will clear (initialize) all devices on the bus that have a device clear function, whether or not the controller has addressed them.
- Group Execute Trigger (GET). Triggers all applicable devices and causes them to initiate their programmed actions.
- Go To Local (GTL). Causes the listen-addressed device to switch from remote to local (front-panel) control.
- Local Lockout (LLO). Disables the return to local function.
- Parallel Poll Configure (PPC). Causes the listen-addressed device to respond to the secondary commands Parallel Poll Enable (PPE) and Parallel Poll Disable (PPD), which are placed on the bus following the PPC command. PPE enables a device with parallel poll capability to respond on a particular data line. PPD disables the device from responding to the parallel poll.
- Select Device Clear (SDC). Clears or initializes all listen-addressed devices.
- Serial Poll Disable (SPD). Changes all devices on the bus from the serial poll state to the normal operating state.
- Serial Poll Enable (SPE). Puts all bus devices that have a service request function into the serial poll enabled state. In this state, each device sends the controller its status byte, instead of its normal output, after the device receives its talk address on the data lines. This function may be used to determine which device sent a service request.
- Take Control (TCT). Allows the controller in charge to pass control of the bus to another controller on the bus.

## Appendix C: Network Interface Specification

The waveform generator supports remote control with the Ethernet interface. This Appendix describes the network interface specification.

TCP/IP is used as the network protocol, and the port number is fixed as 4000. Commands can be sent from the application program through the TCP/IP socket interface, and queries can be received through the interface.

The following lists the differences between the GPIB interface and the Ethernet interface.

- The Line Feed (LF) code is needed as a terminator at the end of a message.
- The IEEE 488.1 standard (for instance, Device Clear, Service Request, etc.) is not supported.
- The Message Exchange Control Protocol in the IEEE 488.2 is not supported. However, common commands such as \*ESE and the event handling features are supported.
- The Indefinite format (the block start at #0) in the <ARBITRARY BLOCK PROGRAM DATA> of the IEEE 488.2 is not supported.

For detailed information about the programming, refer to the *Sample Programs* disk supplied with this waveform generator.



# Appendix D: SCPI Conformance Information

All commands in the waveform generator are based on SCPI Version 1999.0 (AWG400 series) or SCPI Version 1995.0 (AWG500/600 series). Table D–1 lists the SCPI commands this waveform generator supports.

**Table D–1: SCPI conformance information(AWG400 series)**

Command	Defined in SCPI 1999.0	Not defined in SCPI 1999.0
ABORt	✓	
ABSTouch		✓
AWGcontrol		
DOUTput	[STATe] (?)	✓
ENHanced	SEQuence [JMODE] (?)	✓
EVENT	LOGic [IMMediate]	✓
	SOFTware [IMMediate]	✓
	TABLE [IMMediate]	✓
FG	FREQuency [CW FIXed] (?)	✓
	FUNCTion [SHAPE] (?)	✓
	PHASe [ADJust] (?)	✓
	POLarity (?)	✓
	PULSe DCYCLE (?)	✓
	[STATe] (?)	✓
	VOLTage [LEVe] [IMMediate] [AMPLitude] (?)	✓
	VOLTage [LEVe] [IMMediate] OFFSet(?)	✓
	RMODE(?)	✓
	RSTATe?	✓
	RUN [IMMediate]	✓
	SREStore	✓
	SSAVe	✓
	STOP [IMMediate]	✓
CALibration	[ALL] (?)	✓
DIAGnostic	DATA?	✓
	[IMMediate] (?)	✓
	SElect(?)	✓
DISPlay	BRIGHtness(?)	✓
	ENABLe(?)	✓

**Table D-1: SCPI conformance information(AWG400 series) (Cont.)**

Command				Defined in SCPI 1999.0	Not defined in SCPI 1999.0	
HCOPY	HILight	COLor(?)			✓	
	DESTination			✓		
	DEVice	COLor(?)		✓		
		LANGuage(?)		✓		
	[IMMediate]			✓		
SDUMp	[IMMediate]		✓			
MMEMory	CATalog?			✓		
	CDIRectory(?)			✓		
	CLOSE			✓		
	COPY			✓		
	DATA			✓		
	DElete			✓		
	FEED(?)			✓		
	INITialize			✓		
	MDIRectory			✓		
	MSIS(?)				✓	
	MOVE			✓		
	NAME(?)			✓		
	OPEN			✓		
	OUTPut	FILTer	[LPASs]	FREQuency(?)	✓	
ISTate(?)					✓	
[STATe](?)				✓		
SOURce	COMBine	FEED (?)		✓		
	FREQuency	[CW FIXed] (?)		✓		
	FUNction	USER(?)			✓	
	MARKer	DElay(?)				✓
		VOLTage	[LEVe1]	[IMMediate] HIGH(?)		✓
		VOLTage	[LEVe1]	[IMMediate] LOW(?)		✓
	POWer	[LEVe1]	[IMMediate]	[AMPLitude] (?)	✓	
	ROSCillator	SOURce(?)		✓		
	SKEW(?)					✓
	VOLTage	[LEVe1]	[IMMediate]	[AMPLitude] (?)	✓	
OFFSet (?)				✓		
HIGH(?)				✓		

Table D-1: SCPI conformance information(AWG400 series) (Cont.)

Command		Defined in SCPI 1999.0	Not defined in SCPI 1999.0		
	LOW(?)	✓			
STATus	OPERation	[EVENT]?	✓		
		CONDition?	✓		
		ENABle(?)	✓		
	PRESet	✓			
	QUESTionable	[EVENT]?	✓		
		CONDition?	✓		
ENABle(?)		✓			
SYSTem	BEEPer	[IMMediate]	✓		
	COMMunicate	LAN DHCP [CLIent] LEASe TIME(?)		✓	
		LAN DHCP [CLIent] [STATe] (?)		✓	
		LAN FTP [SERVer] [STATe] (?)		✓	
		VERSion(?)		✓	
		GATeway	ADDRes(?)	✓	
		NFS	TLIMit(?)	✓	
		PING?		✓	
		RDEvice	ADDRes(?)		✓
			FSYSstem(?)		✓
			NAME(?)		✓
			PROTOcol (?)		✓
			[STATe] (?)		✓
		[SELF]	ADDRes(?)		✓
	MADDress(?)		✓		
	SMASk(?)		✓		
	DATE(?)		✓		
	ERRor	[NEXT]?	✓		
	KDIRrection(?)			✓	
	KEYBoard	[TYPE] (?)		✓	
KLOCK(?)		✓			
SECurity	IMMediate	✓			
TIME(?)		✓			
UPTime?			✓		
VERSion?		✓			

**Table D-2: SCPI conformance information(AWG500/600 series)**

Command		Defined in SCPI 1995.0	Not defined in SCPI 1995.0	
ABORt		✓		
ABSTouch			✓	
AWGcontrol	DOUTput [STATe] (?)		✓	
	ENHanced SEQuence [JMODE] (?)		✓	
	EVENT	LOGic [IMMediate]		✓
		SOFTware [IMMediate]		✓
		TABLe [IMMediate]		✓
	FG	FREQuency [CW FIXed] (?)		✓
		FUNCTion [SHAPE] (?)		✓
		PHASe [ADJust] (?)		✓
		POLarity (?)		✓
		PULSe DCYCLe (?)		✓
		[STATe] (?)		✓
		VOLTage [LEVeL] [IMMediate] [AMPLitude] (?)		✓
	VOLTage [LEVeL] [IMMediate] OFFSet (?)		✓	
	RMODE(?)			✓
	RSTATE?			✓
	RUN [IMMediate]		✓	
	SREStore			✓
SSAVe			✓	
STOP [IMMediate]			✓	
CALibration [ALL] (?)	✓			
DIAGnostic	DATA?		✓	
	[IMMediate] (?)		✓	
	SElect (?)		✓	
DISPlay	BRIGhtness (?)	✓		
	ENABLe (?)	✓		
	HILight COLor (?)		✓	



Table D-2: SCPI conformance information(AWG500/600 series) (Cont.)

Command		Defined in SCPI 1995.0	Not defined in SCPI 1995.0	
HCOPY	DESTination	✓		
	DEvice	COLor(?)	✓	
		LANGuage(?)	✓	
	[IMMediate]		✓	
	SDUMp	[IMMediate]	✓	
MMEMory	CATalog?	✓		
	CDIRectory(?)	✓		
	CLOSe	✓		
	COPY	✓		
	DATA	✓		
	DELeTe	✓		
	FEED(?)	✓		
	INITialize	✓		
	MDIRectory	✓		
	MSIS(?)		✓	
	MOVE	✓		
	NAME(?)	✓		
	OPEN	✓		
	OUTPut	FILTer	[LPASS] FREQuency(?)	✓
ISTate(?)			✓	
[STATe](?)			✓	
SOURce	COMBine	FEED (?)	✓	
	FREQuency	[CW FIXed](?)	✓	
	FUNCTion	USER(?)		✓
	MARKer	DELaY(?)		✓
		VOLTage	[LEVe1] [IMMediate] HIGH(?)	✓
		VOLTage	[LEVe1] [IMMediate] LOW(?)	✓
	POWer	[LEVe1] [IMMediate] [AMPLitude](?)	✓	
	ROSCillator	SOURce(?)	✓	
	SKEW(?)			✓
	VOLTage	[LEVe1] [IMMediate] [AMPLitude](?)	✓	
OFFSet(?)		✓		
HIGH(?)		✓		
LOW(?)		✓		

**Table D-2: SCPI conformance information(AWG500/600 series) (Cont.)**

Command		Defined in SCPI 1995.0	Not defined in SCPI 1995.0		
STATus	OPERation	[EVENT]?	✓		
		CONDition?	✓		
		ENABle(?)	✓		
	PRESet	✓			
	QUEStionable	[EVENT]?	✓		
		CONDition?	✓		
		ENABle(?)	✓		
QUEue	[NEXT] ?	✓			
SYSTem	BEEPer	[IMMediate]	✓		
	COMMunicate	LAN DHCP [CLient] LEASe TIME(?)		✓	
		LAN DHCP [CLient] [STATe] (?)		✓	
		LAN FTP [SERVer] [STATe] (?)		✓	
		VERSION(?)		✓	
		GATeway ADDRESS(?)		✓	
		NFS TLIMit(?)		✓	
		PING?		✓	
		RDEVice	ADDRESS(?)		✓
			FSYSTem(?)		✓
			NAME(?)		✓
			PROTOcol (?)		✓
			[STATe] (?)		✓
		[SELF]	ADDRESS(?)		✓
	MADDRESS(?)		✓		
	SMASK(?)		✓		
	DATE(?)	✓			
	ERRor?	✓			
	KDIRectio n(?)		✓		
	KEYBoard [TYPE] (?)		✓		
KLOCK(?)	✓				
SECurity IMMEDIATE	✓				
TIME(?)	✓				
UPTime?		✓			
VERSion?	✓				

# Appendix E: Factory Initialization Settings

The following tables lists the commands affected by factory initialization. Table E-1 on page E-1 lists commands for the AWG400 series, and Table E-2 on page E-4 lists commands for the AWG500/600 series.

The SYSTem:SECurity:IMMediate command initializes all the settings as shown below; the \*RST command has no effect on the Status commands and the SYSTem:COMMunicate:LAN commands.

**Table E-1: Factory initialization settings (AWG400 series)**

Header	Default settings
<b>AWGcontrol commands</b>	
AWGControl:CLOCK:SOURce	INTernal
AWGControl:DOUtpuT[1 2]	0
AWGControl:ENHanced:SEQuence[:JMODE]	TABLe
AWGControl:FG:FREQuency[:CW :FIXed]	1.0MHz
AWGControl:FG<n>:FUNctIon[:SHAPE]	SINusoid
AWGControl:FG<n>:PHASe[:ADJust]	0.000
AWGControl:FG<n>:POLarity	POSitive
AWGControl:FG<n>:PULSe:DCYCLe	10.0
AWGControl:FG[:STATe]	0
AWGControl:FG<n>:VOLTagE[:LEVeL] [:IMMediate] [:AMPLitude]	1.000
AWGControl:FG<n>:VOLTagE[:LEVeL] [:IMMediate] :OFFSet	0.000
AWGControl:RMODE	CONTInuous
<b>Diagnostic commands</b>	
DIAGnostic:SElect	ALL
<b>Display commands</b>	
DISPlay:ENABle	1
DISPlay:HILight:COLor	0
<b>Hardcopy commands</b>	
HCOPy:DEVice:COLor	0
HCOPy:DEVice:LANGuage	BMP

**Table E-1: Factory initialization settings (AWG400 series) (Cont.)**

<b>Header</b>	<b>Default settings</b>
<b>MMemory commands</b>	
MMEMory:CDIRectory	"/"
MMEMory:FEED	"HCOP"
MMEMory:MSIS	"MAIN"
MMEMory:NAME	"HARDCOPY", "MAIN"
<b>Output commands</b>	
OUTPut[1 2 3]:FILTer[:LPASs]:FREQuency	9.9E+37
OUTPut[1 2 3]:ISTate	0
OUTPut[1 2 4 5 6][:STATe]	0
<b>Source commands</b>	
[SOURce[1 2 3]]:COMBine:FEED	"" (null)
[SOURce[1 2 3 4 5 6]]:FREQuency[:CW]:FIXed	1.0000000E+8
[SOURce[1 2 3 4 5 6]]:FUNCTion:USER	"" (null), "MAIN"
SOURce[7 9 11]:POWer[:LEVe1][:IMMediate][:AMPLitude]	-95
[SOURce[1 2 3 4 5 6]]:ROSCillator:SOURce	INTerna1
SOURce[2 3 5 6]:SKEW	0.0
[SOURce[1 2]]:VOLTage[:LEVe1][:IMMediate][:AMPLitude]	1.000
[SOURce[1 2]:]VOLTage[:LEVe1][:IMMediate]:OFFSet	0.000
<b>Status commands</b>	
*ESE <sup>1</sup>	0
*PSC <sup>1</sup>	1
*SRE <sup>1</sup>	0
STATus:OPERation:ENABle <sup>1</sup>	0
STATus:QUESTionable:ENABle <sup>1</sup>	0

**Table E-1: Factory initialization settings (AWG400 series) (Cont.)**

Header	Default settings
<b>System commands</b>	
SYSTem:COMMunicate:LAN:DHCP[:CLient]:LEASE:TIME	28800
SYSTem:COMMunicate:LAN:DHCP[:CLient][:STATE] <sup>1</sup>	0
SYSTem:COMMunicate:LAN:FTP[:SERVer][:STATE] <sup>1</sup>	0
SYSTem:COMMunicate:LAN:FTP[:SERVer]:VERSION <sup>1</sup>	STANdard
SYSTem:COMMunicate:LAN:GATeway<x>:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:NFS:TLMit	300
SYSTem:COMMunicate:LAN:RDEvice<x>:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:RDEvice<x>:FSYSem <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:RDEvice<x>:NAME <sup>1</sup>	"NET<x>"
SYSTem:COMMunicate:LAN:RDEvice<x>:PROTocol <sup>1</sup>	NFS
SYSTem:COMMunicate:LAN:RDEvice<x>[:STATE] <sup>1</sup>	0
SYSTem:COMMunicate:LAN[:SELF]:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN[:SELF]:SMASK <sup>1</sup>	"" (null)
SYSTem:KDIRectioN	FORWard
SYSTem:KEYBoard[:TYPE]	ASCii
SYSTem:KLOCK	0
<b>Trigger commands</b>	
TRIGger[:SEquence]:IMPedance	5.0E+1
TRIGger[:SEquence]:LEVel	1.4
TRIGger[:SEquence]:POLarity	POSitive
TRIGger[:SEquence]:SLOPe	POSitive
TRIGger[:SEquence]:SOURce	EXTernal
TRIGger[:SEquence]:TIMER	1.00E-1

<sup>1</sup> These commands are not affected by the \*RST command.

The following table lists the AWG 500/600 series commands affected by factory initialization. Table E-1 on page E-1 lists affected commands for the AWG400 series.

**Table E-2: Factory initialization settings (AWG500/600 series)**

<b>Header</b>	<b>Default settings</b>
<b>AWGcontrol commands</b>	
AWGControl:CLOCK:SOURce	INTernal (AWG500 series only)
AWGControl:DOUTput[1 2]	0
AWGControl:ENHanced:SEQUence[:JMODE]	TABLE
AWGControl:FG:FREQuency[:CW :FIXed]	5.0MHz (AWG500 series) 13.0MHz (AWG600 series)
AWGControl:FG<n>:FUNctIon[:SHAPE]	SINusoid
AWGControl:FG<n>:PHASe[:ADJust]	0.000
AWGControl:FG<n>:POLarity	POSitive
AWGControl:FG<n>:PULSe:DCYCLe	10.0
AWGControl:FG[:STATE]	0
AWGControl:FG<n>:VOLTage[:LEVe1][:IMMediate][:AMPLitude]	1.000
AWGControl:FG<n>:VOLTage[:LEVe1][:IMMediate]:OFFSet	0.000
AWGControl:RMODE	CONTInuous
<b>Diagnostic commands</b>	
DIAGnostic:SElect	ALL
<b>Display commands</b>	
DISPly:BRIGhtness	0.7
<b>Hardcopy commands</b>	
HCOPY:DEVIce:LANGuage	BMP
<b>MMemory commands</b>	
MMEMory:CDIRectory	"/"
MMEMory:FEED	"HCOP"
MMEMory:MSIS	"MAIN"
MMEMory:NAME	"HARDCOPY", "MAIN"

**Table E-2: Factory initialization settings (AWG500/600 series) (Cont.)**

<b>Header</b>	<b>Default settings</b>
<b>Output commands</b>	
OUTPut[1 2]:FILTer[:LPASS]:FREQuency	9.9E+37
OUTPut[1]:ISTate	0
OUTPut[1 2 5 7][:STATe]	0
<b>Source commands</b>	
[SOURce1]:COMBine:FEED	"" (null)
[SOURce[1 2 5]]:FREQuency[:CW]:FIXed]	1.0000000E+8
[SOURce[1 2 5]]:FUNction:USER	"" (null), "MAIN"
[SOURce[1 2 5]]:MARKer<y>:DELay	0
[SOURce[1 2 5]]:MARKer<y>:VOLTagE[:LEVe1] [:IMMediate]:HIGH	2.00
[SOURce[1 2 5]]:MARKer<y>:VOLTagE[:LEVe1] [:IMMediate]:LOW	0.00
SOURce7:POWer[:LEVe1][:IMMediate][:AMPLitude]	-105
[SOURce[1 2 5]]:ROScillator:SOURce	INTernal
[SOURce[1 2]]:VOLTagE[:LEVe1][:IMMediate] [:AMPLitude]	1.000
SOURce5:VOLTagE[:LEVe1][:IMMediate]:HIGH	2.00
SOURce5:VOLTagE[:LEVe1][:IMMediate]:LOW	0.00
[SOURce[1 2]:]VOLTagE[:LEVe1][:IMMediate]: OFFSet	0.000
<b>Status commands</b>	
*ESE <sup>1</sup>	0
*PSC <sup>1</sup>	1
*SRE <sup>1</sup>	0
STATus:OPERation:ENABle <sup>1</sup>	0
STATus:QUESTionable:ENABle <sup>1</sup>	0

**Table E-2: Factory initialization settings (AWG500/600 series) (Cont.)**

Header	Default settings
<b>System commands</b>	
SYSTem:COMMunicate:LAN:FTP[:SERVer][:STATe] <sup>1</sup>	0
SYSTem:COMMunicate:LAN:FTP[:SERVer]:VERSion <sup>1</sup>	STANdard
SYSTem:COMMunicate:LAN:GATEeway<x>:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:NFS:TLIMit	300
SYSTem:COMMunicate:LAN:RDEVice<x>:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:RDEVice<x>:FSYStem <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN:RDEVice<x>:NAME <sup>1</sup>	"NET<x>"
SYSTem:COMMunicate:LAN:RDEVice<x>:PROTOcol <sup>1</sup>	NFS
SYSTem:COMMunicate:LAN:RDEVice<x>[:STATe] <sup>1</sup>	0
SYSTem:COMMunicate:LAN[:SELF]:ADDRess <sup>1</sup>	"" (null)
SYSTem:COMMunicate:LAN[:SELF]:SMASK <sup>1</sup>	"" (null)
SYSTem:KDIRectioN	FORWard
SYSTem:KEYBoArD[:TYPE]	ASCii
SYSTem:KLOCK	0
<b>Trigger commands</b>	
TRIGger[:SEQuence]:IMPedance	1.0E+03
TRIGger[:SEQuence]:LEVel	1.4
TRIGger[:SEQuence]:POLarity	POSitive
TRIGger[:SEQuence]:SLOPe	POSitive
TRIGger[:SEQuence]:SOURce	EXTernal
TRIGger[:SEQuence]:TIMer	1.00E-1

**1** These commands are not affected by the \*RST command.





# **Glossary and Index**



# Glossary

**ASCII**

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

**Address**

A 7-bit code that identifies an instrument on the communication bus. The instrument must have a unique address for the controller to recognize and transmit commands.

**BNF (Backus-Naur Form)**

A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

**Controller**

A computer or other device that sends commands to and accepts responses from the digitizing oscilloscope.

**EOI**

A mnemonic referring to the control line “End or Identify” on the GPIB interface bus. One of the two possible end-of-message terminators.

**EOM**

A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

**GPIB**

Acronym for General Purpose Interface Bus, the common name for the communications interface system defined in IEEE Std 488.

**IEEE**

Acronym for the Institute for Electrical and Electronic Engineers.

**QuickC**

A computer language (distributed by Microsoft) that is based on C.

**SCPI**

Acronym for Standard Commands for Programmable Instruments.



# Index

## A

Abbreviations, commands, queries, and parameters 2-5  
ABORt 2-31  
ABSTouch 2-32  
Arguments, parameters 2-4  
AWG control command group 2-20  
AWG control commands  
  AWGControl:CLOCK:SOURce 2-37  
  AWGControl:DOUtpuT<x>[:STATe] 2-37  
  AWGControl:ENHanced:SEQuence[:JMODE] 2-39  
  AWGControl:EVENT[:LOGic][:IMMediate] 2-39  
  AWGControl:EVENT:SOFTware[:IMMediate] 2-40  
  AWGControl:EVENT:TABLE[:IMMediate] 2-41  
  AWGControl:FG:FREQuency[:CW|:FIXed] 2-41  
  AWGControl:FG<n>:FUNCTion[:SHAPE] 2-42  
  AWGControl:FG<n>:PHASe[:ADJust] 2-44  
  AWGControl:FG<n>:POLarity 2-45  
  AWGControl:FG<n>:PULSe:DCYCLE 2-46  
  AWGControl:FG[:STATe] 2-47  
  AWGControl:FG<n>:VOLTagE[:LEVel][:IMMediate][:AMPLitude] 2-48  
  AWGControl:FG<n>:VOLTagE[:LEVel][:IMMediate]:OFFSet 2-49  
  AWGControl:RMODE 2-50  
  AWGControl:RSate? 2-51  
  AWGControl:RUN[:IMMediate] 2-51  
  AWGControl:SREStore 2-52  
  AWGControl:SSAVe 2-53  
  AWGControl:STOP[:IMMediate] 2-53  
AWGControl:CLOCK:SOURce 2-37  
AWGControl:DOUtpuT<x>[:STATe] 2-37  
AWGControl:ENHanced:SEQuence[:JMODE] 2-39  
AWGControl:EVENT[:LOGic][:IMMediate] 2-39  
AWGControl:EVENT:SOFTware[:IMMediate] 2-40  
AWGControl:EVENT:TABLE[:IMMediate] 2-41  
AWGControl:FG:FREQuency[:CW|:FIXed] 2-41  
AWGControl:FG<n>:FUNCTion[:SHAPE] 2-42

AWGControl:FG<n>:PHASe[:ADJust] 2-44  
AWGControl:FG<n>:POLarity 2-45  
AWGControl:FG<n>:PULSe:DCYCLE 2-46  
AWGControl:FG[:STATe] 2-47  
AWGControl:FG<n>:VOLTagE[:LEVel][:IMMediate][:AMPLitude] 2-48  
AWGControl:FG<n>:VOLTagE[:LEVel][:IMMediate]:OFFSet 2-49  
AWGControl:RMODE 2-50  
AWGControl:RSate? 2-51  
AWGControl:RUN[:IMMediate] 2-51  
AWGControl:SREStore 2-52  
AWGControl:SSAVe 2-53  
AWGControl:STOP[:IMMediate] 2-53

## B

Backus–Naur Form 2-1  
BNF (Backus–Naur form) 2-1

## C

\*CAL? 2-54  
Calibration command group 2-21  
Calibration commands  
  \*CAL? 2-54  
  CALibration[:ALL] 2-55  
CALibration[:ALL] 2-55  
Case sensitivity 2-8  
Character chart A-1  
\*CLS 2-56  
Code, error 3-15  
Code Convert File 2-137  
Command, summaries 2-20  
Command group  
  AWG control 2-20  
  Calibration 2-21  
  Diagnostic 2-21  
  Display 2-21  
  Hardcopy 2-22  
  Mass memory 2-23  
  Output 2-24  
  Source 2-25  
  Status 2-27  
  Synchronization 2-27  
  System 2-28  
  Trigger 2-30  
Command Groups 2-17  
  Command Quick Reference 2-18  
  Function Groups 2-17  
Command Quick Reference 2-18  
Command syntax 2-1

Commands  
  chaining 2–6  
  Parts of 1–1  
  structure of IEEE 488.2 commands 2–9  
Constructed Mnemonics 2–10  
Creating commands 2–3

## D

Data File 2–129  
Data Transfer, Data File 2–129, 2–130  
Data transfer 2–129  
DCL B–3  
Device Clear B–3  
Diagnostic command group 2–21  
Diagnostic commands  
  \*TST? 2–124  
  DIAGnostic:DATA? 2–56  
  DIAGnostic[:IMMediate] 2–57  
  DIAGnostic:SElect 2–58  
DIAGnostic:DATA? 2–56  
DIAGnostic[:IMMediate] 2–57  
DIAGnostic:SElect 2–58  
Diagram, syntax 2–15  
Disks included with this manual 1–3  
Display command group 2–21  
Display commands  
  ABSTouch 2–32  
  DISPlay:BRIGhtness 2–59  
  DISPlay:ENABle 2–59  
  DISPlay:HILight:COLor 2–60  
DISPlay:BRIGhtness 2–59  
DISPlay:ENABle 2–59  
DISPlay:HILight:COLor 2–60

## E

Enable registers 3–8  
equation file 2–136  
Error code 3–15  
Error codes  
  AWG400/500/600 unique 3–23  
  commands 3–16  
  device specific 3–20  
  execution 3–18  
  hardware 3–20, 3–23  
  query 3–21  
Error message 3–15  
\*ESE 2–61  
\*ESR 2–62  
Ethernet  
  Network test 1–10  
  Setting parameters 1–9  
Example programs 1–3

## F

Factory initialization settings E–1

File  
  Code Convert 2–137  
  equation 2–136  
  Pattern 2–132  
  Sequence 2–134  
  Waveform 2–131  
Function Groups 2–17

## G

GET B–3  
Go to local B–3  
 GPIB  
  Configurations 1–6  
  Connection rules 1–6  
  interface messages B–3  
  interface specification B–1  
  Setting parameters 1–6  
Group execute trigger B–3  
Groups, command 2–17  
GTL B–3

## H

Hardcopy command group 2–22  
Hardcopy commands  
  HCOPY:DESTination 2–62  
  HCOPY:DEvice:COLor 2–63  
  HCOPY:DEvice:LANGuage 2–63  
  HCOPY:SDUMp[:IMMediate] 2–65  
  HCOPY[:IMMediate] 2–64  
HCOPY:DESTination 2–62  
HCOPY:DEvice:COLor 2–63  
HCOPY:DEvice:LANGuage 2–63  
HCOPY:SDUMp[:IMMediate] 2–65  
HCOPY[:IMMediate] 2–64

## I

\*IDN? 2–65  
IEEE 488.2 common commands 2–9  
IEEE Std 488.2–1987 1–4  
Instrument setup 1–4, 1–8  
Interface message B–3

## L

LLO B–3  
Local lock out B–3

## M

Mass memory command group 2–23  
Mass memory commands  
  MMEMory:CATalog? 2–66  
  MMEMory:CDIRectory 2–67  
  MMEMory:CLOSe 2–67  
  MMEMory:COpy 2–68  
  MMEMory:DATA 2–69

MMEMory:DELeTe 2-69  
 MMEMory:FEED 2-70  
 MMEMory:INITialize 2-71  
 MMEMory:MDIRectory 2-72  
 MMEMory:MOVE 2-72  
 MMEMory:MSIS 2-73  
 MMEMory:NAME 2-74  
 MMEMory:OPEN 2-75  
 Message, error 3-15  
 MMEMory:CATalog? 2-66  
 MMEMory:CDIRectory 2-67  
 MMEMory:CLOSe 2-67  
 MMEMory:COpy 2-68  
 MMEMory:DATA 2-69  
 MMEMory:DELeTe 2-69  
 MMEMory:FEED 2-70  
 MMEMory:INITialize 2-71  
 MMEMory:MDIRectory 2-72  
 MMEMory:MOVE 2-72  
 MMEMory:MSIS 2-73  
 MMEMory:NAME 2-74  
 MMEMory:OPEN 2-75  
 Mnemonics, Constructed 2-10

## N

National Instruments 4-1  
 Network, interface specification *C-1*  
 Network test, Ethernet 1-10

## O

\*OPC 2-76  
 \*OPT? 2-76  
 Output command group 2-24  
 Output commands  
   OUTPut<x>:FILTer[:LPASs]:FREQUency  
     2-77  
   OUTPut<x>:ISTate 2-78  
   OUTPut<x>[:STATe] 2-79  
 OUTPut<x>:FILTer[:LPASs]:FREQUency 2-77  
 OUTPut<x>:ISTate 2-78  
 OUTPut<x>[:STATe] 2-79

## P

Parallel poll B-3  
 Parameter setting  
   Ethernet 1-9  
   GPIO 1-6  
 Parameter types used in syntax descriptions 2-4  
 Parts of commands 1-1  
 Pattern File 2-132  
 PPC B-3  
 PPD B-3  
 PPE B-3  
 PPU B-3  
 \*PSC 2-80

## Q

Queries 2-3  
 Query Responses 2-3  
 Queues 3-10  
   event 3-10  
   output 3-10  
 Quotes 2-8

## R

README.TXT 4-1  
 Registers 3-4  
   Event Status Enable Register (ESER) 3-8  
   Operation Condition Register (OCR) 3-7  
   Operation Enable Register (OENR) 3-9  
   Operation Event Register (OEVR) 3-7  
   Questionable Condition Register (OCR) 3-7  
   Questionable Enable Register (QENR) 3-9  
   Questionable Event Register (OEVR) 3-8  
   Service Request Enable Register (SRER) 3-8  
   Standard Event Status Register (SESR) 3-6  
   Status Byte Register (SRB) 3-5  
 Response, Retrieving 2-127  
 Retrieving Response 2-127  
 \*RST 2-81  
 Rules, for using SCPI commands 2-8

## S

SCPI  
   abbreviating 2-5  
   chaining commands 2-6  
   commands 2-2  
   conformation information D-1  
   general rules 2-8  
   parameter types 2-4  
   subsystem hierarchy tree 2-2  
 SCPI commands and queries syntax 2-2-2-8  
   creating commands 2-3  
   creating queries 2-3  
 SDC B-3  
 Selected device clear B-3  
 Sequence File 2-134  
 Serial poll  
   Disable B-3  
   Enable B-3  
 Setup, Instrument preparation 1-4, 1-8  
 SI prefix and unit 2-7  
 Source command group 2-25  
 Source commands  
   [SOURce<x>]:COMBine:FEED 2-81  
   [SOURce<x>]:FREQUency[:CW|FIXed]  
     2-83  
   [SOURce<x>]:FUNction:USER 2-84  
   [SOURce<x>]:MARKer[1|2]:DELay 2-85

- [SOURCE<x>]:MARKER[1|2]:VOLTage[:LEV-  
el][:IMMEDIATE]:HIGH 2-86
- [SOURCE<x>]:MARKER[1|2]:VOLTage[:LEV-  
el][:IMMEDIATE]:LOW 2-87
- SOURCE<x>:POWer[:LEVel][:IMMEDIATE]:  
AMPlitude] 2-88
- [SOURCE<x>]:ROSCillator:SOURce 2-89
- SOURCE<x>:SKEW 2-90
- [SOURCE<x>]:VOLTage[:LEVel][:IMMEDIATE]:  
AMPlitude] 2-90
- SOURCE5:VOLTage[:LEVel][:IMMEDIATE]:  
HIGH 2-91
- SOURCE5:VOLTage[:LEVel][:IMMEDIATE]:  
LOW 2-92
- [SOURCE<x>]:VOLTage[:LEVel][:IMMEDIATE]:  
OFFSet 2-93
- [SOURCE<x>]:COMBine:FEED 2-81
- [SOURCE<x>]:FREQuency[:CW|FIXed] 2-83
- [SOURCE<x>]:FUNction:USER 2-84
- [SOURCE<x>]:MARKER[1|2]:DELay 2-85
- [SOURCE<x>]:MARKER[1|2]:VOLTage[:LEV-  
el][:IMMEDIATE]:HIGH 2-86
- [SOURCE<x>]:MARKER[1|2]:VOLTage[:LEV-  
el][:IMMEDIATE]:LOW 2-87
- SOURCE<x>:POWer[:LEVel][:IMMEDIATE]:AM-  
Plitude] 2-88
- [SOURCE<x>]:ROSCillator:SOURce 2-89
- SOURCE<x>:SKEW 2-90
- [SOURCE<x>]:VOLTage[:LEVel][:IMMEDIATE]:  
AMPlitude] 2-90
- SOURCE5:VOLTage[:LEVel][:IMMEDIATE]:HIGH  
2-91
- SOURCE5:VOLTage[:LEVel][:IMMEDIATE]:LOW  
2-92
- [SOURCE<x>]:VOLTage[:LEVel][:IMMEDIATE]:  
OFFSet 2-93
- SPD B-3
- SPE B-3
- Special characters 2-5
- \*SRE 2-94
- Status and error commands
  - STATUS:OPERation:ENABLE 3-9
  - STATUS:QUEStionable:ENABLE 3-9
- Status and events, displaying on screen 3-13
- Status command group 2-27
- Status commands
  - \*CLS 2-56
  - \*ESE 2-61
  - \*ESR? 2-62
  - \*PSC 2-80
  - \*SRE 2-94
  - \*STB? 2-100
  - STATUS:OPERation:CONDition? 2-95
  - STATUS:OPERation:ENABLE 2-96
  - STATUS:OPERation[:EVENT]? 2-96
  - STATUS:PRESet 2-97
  - STATUS:QUEStionable:CONDition? 2-97
  - STATUS:QUEStionable:ENABLE 2-98
  - STATUS:QUEStionable[:EVENT]? 2-99
  - STATUS:QUEue[:NEXT]? 2-99
- Status registers 3-4
- Status reporting 3-1
- STATUS:OPERation:CONDition? 2-95
- STATUS:OPERation:ENABLE 2-96, 3-9
- STATUS:OPERation[:EVENT]? 2-96
- STATUS:PRESet 2-97
- STATUS:QUEStionable:CONDition? 2-97
- STATUS:QUEStionable:ENABLE 2-98, 3-9
- STATUS:QUEStionable[:EVENT]? 2-99
- STATUS:QUEue[:NEXT]? 2-99
- \*STB? 2-100
- Synchronization command group 2-27
- Synchronization commands
  - \*OPC 2-76
  - \*WAI 2-125
- Synchronizing execution 3-14
- Syntax
  - Command 2-1
  - diagrams 2-15
- Syntax diagrams 1-1
- System command group 2-28
- System commands
  - \*IDN? 2-65
  - \*OPT? 2-76
  - \*RST 2-81
  - SYSTEM:BEEPer[:IMMEDIATE] 2-101
  - SYSTEM:COMMunicate:LAN:DHCP[:CLI-  
ent]:LEASe:TIME 2-101
  - SYSTEM:COMMunicate:LAN:DHCP[:CLI-  
ent][:STATe] 2-102
  - SYSTEM:COMMunicate:LAN:FTP[:SERV-  
er][:STATe] 2-103
  - SYSTEM:COMMunicate:LAN:FTP[:SERV-  
er]:VERsion 2-103
  - SYSTEM:COMMunicate:LAN:GATE-  
way[1|2|3]:ADDResS 2-104
  - SYSTEM:COMMunicate:LAN:NFS:TLIMit  
2-105
  - SYSTEM:COMMunicate:LAN:PING? 2-105
  - SYSTEM:COMMunicate:LAN:RDE-  
Vice[1|2|3]:ADDResS 2-106
  - SYSTEM:COMMunicate:LAN:RDE-  
Vice[1|2|3]:FSYSem 2-107
  - SYSTEM:COMMunicate:LAN:RDE-  
Vice[1|2|3]:NAME 2-107
  - SYSTEM:COMMunicate:LAN:RDE-  
Vice[1|2|3]:PROToCol 2-108



SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3][:STATe] 2–109  
 SYSTem:COMMunicate:LAN[:SELF]:AD-  
   DRess 2–109  
 SYSTem:COMMunicate:LAN[:SELF]:MAD-  
   DRess 2–110  
 SYSTem:COMMunicate:LAN[:SELF]:SMASK  
   2–111  
 SYSTem:DATE 2–112  
 SYSTem:ERRor? 2–113  
 SYSTem:ERRor[:NEXT]? 2–114  
 SYSTem:KDIRectioN 2–115  
 SYSTem:KEYBoard[:TYPE] 2–115  
 SYSTem:KLOCK 2–116  
 SYSTem:SECurity:IMMediate 2–117  
 SYSTem:TIME 2–117  
 SYSTem:UPTime? 2–118  
 SYSTem:VERSion? 2–119  
 SYSTem:BEEPer[:IMMediate] 2–101  
 SYSTem:COMMunicate:LAN:DHCP[:CLI-  
   ent]:LEASe:TIME 2–101  
 SYSTem:COMMunicate:LAN:DHCP[:CLI-  
   ent][:STATe] 2–102  
 SYSTem:COMMunicate:LAN:FTP[:SERV-  
   er][:STATe] 2–103  
 SYSTem:COMMunicate:LAN:FTP[:SERV-  
   er]:VERSion 2–103  
 SYSTem:COMMunicate:LAN:GATe-  
   way[1|2|3]:ADDReSS 2–104  
 SYSTem:COMMunicate:LAN:NFS:TLIMit  
   2–105  
 SYSTem:COMMunicate:LAN:PING? 2–105  
 SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3]:ADDReSS 2–106  
 SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3]:FSYSTem 2–107  
 SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3]:NAME 2–107  
 SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3]:PROToCol 2–108  
 SYSTem:COMMunicate:LAN:RDE-  
   Vice[1|2|3][:STATe] 2–109  
 SYSTem:COMMunicate:LAN[:SELF]:ADDReSS  
   2–109  
 SYSTem:COMMunicate:LAN[:SELF]:MADDReSS  
   2–110  
 SYSTem:COMMunicate:LAN[:SELF]:SMASK  
   2–111

SYSTem:DATE 2–112  
 SYSTem:ERRor? 2–113  
 SYSTem:ERRor[:NEXT]? 2–114  
 SYSTem:KDIRectioN 2–115  
 SYSTem:KEYBoard[:TYPE] 2–115  
 SYSTem:KLOCK 2–116  
 SYSTem:SECurity:IMMediate 2–117  
 SYSTem:TIME 2–117  
 SYSTem:UPTime? 2–118  
 SYSTem:VERSion? 2–119

## T

TCT B–3  
 Transfer, data 2–129  
 \*TRG 2–119  
 Trigger command group 2–30  
 Trigger commands  
   \*TRG 2–119  
   ABORt 2–31  
   TRIGger[:SEQuence][:IMMediate] 2–120  
   TRIGger[:SEQuence]:IMPedance 2–120  
   TRIGger[:SEQuence]:LEVel 2–121  
   TRIGger[:SEQuence]:POLarity 2–121  
   TRIGger[:SEQuence]:SLOPe 2–122  
   TRIGger[:SEQuence]:SOURce 2–123  
   TRIGger[:SEQuence]:TIMer 2–123  
 TRIGger[:SEQuence][:IMMediate] 2–120  
 TRIGger[:SEQuence]:IMPedance 2–120  
 TRIGger[:SEQuence]:LEVel 2–121  
 TRIGger[:SEQuence]:POLarity 2–121  
 TRIGger[:SEQuence]:SLOPe 2–122  
 TRIGger[:SEQuence]:SOURce 2–123  
 TRIGger[:SEQuence]:TIMer 2–123  
 \*TST? 2–124

## U

Unit and SI prefix 2–7  
 UNL B–3  
 Unlisten B–3  
 UNT B–3  
 Untalk B–3

## W

\*WAI 2–125  
 Waveform and Pattern Files 2–130  
 Waveform File 2–131  
 Where to find other information ix





